PROPOSAL

to potential shareholders
in a corporation to develop and market computer software.

Prepared by Rick Glasel <rick.glasel@sloughware.com> on October 28th, 2009.

Please refer to the final page of this document for disclaimers and other important notes

What is Mallard?

Mallard is the code name for a computer program that will make it easier to manage all of the information related to our daily activities, in our work or professional lives as well as our personal lives. The information Mallard manages could be our schedules, addresses and phone numbers for people we know, information about our customers, our relatives, virtually anything worth writing down on a piece of paper (because we cannot take a chance on forgetting it).

The program itself will be freely available, but it will only work with personalized and encrypted databases. Mallard databases can only be opened by someone who has purchased a Mallard license with the key to decrypt that database. Individual licenses will sell for between $30 and $50, and there are profitable opportunities to sell information services that are integrated with Mallard, and to sell software and services to companies so they can operate servers that allow for the secure exchange of information between Mallard databases.

Who are the potential customers for Mallard?

This is somewhat of a classic Catch 22 situation. To successfully sell a program that can be used for both work and personal purposes, the program must be acceptable to employers. Employers are reluctant to adopt a new program unless it is already successful in the marketplace. The only way to break into the business market is to build a good reputation with individual users first, so that Mallard can be promoted as a proven product.

The first people to be targeted for Mallard will be already using a program like Microsoft Outlook or Google Calendar to keep their schedules and contact information better organized and closer at hand, compared to a pile of Post-It notes and a calendar tacked on the wall. The demands on their time are such that they have to juggle their personal and work schedules, but they are uncomfortable with sharing their personal calendar with their employer. They do not want to spend any more time than necessary typing their information into a computer. They use computers on a daily basis, mostly because they need to, not because that is their hobby or vocation. Their time is worth something, so they will consider paying for a program that actually saves them time and makes their life a little easier, but there has to be a noticeable difference between that program and a free program.

The main competitors to Mallard are either free, included with Microsoft Office Professional, or part of a bigger software package paid for by employers. This type of software has not changed significantly in fifteen years. Word of mouth promotion, a professionally designed website, and a product that knocks people's socks off is at least 90% of the marketing that can be done with Mallard. Word of mouth promotion will start with free licenses given to dozens of people who agree to become "beta" testers. Beta testers agree to provide feedback on programs that are close to being ready for sale to the public. One reason to use beta testers is to trap unexpected bugs in a program before it is available for sale, another reason is to use those testers as

“character” references for the program. To emphasize the importance of these beta testers (along with the first paying customers), their feedback will strongly influence the final design of Mallard, and help to determine which features are added and enhanced once the program is for sale. Updates to the program itself will always be free.

Development of the Mallard program, and services to go with the program, will not stop once the program is released to the public. Development of a version aimed at professionals working together in a clinical setting will start as soon as the general version for individuals is ready for the marketplace. There is great potential for information services that can be integrated with a Mallard database and provided over the Internet, in fact, this could eventually become a key selling feature. As the number of individuals using Mallard grows, the potential for sales to businesses also grows, and every time a business decides to use Mallard, the number of individuals using Mallard will grow. Getting the first version of Mallard to market should be the beginning of a successful long-term business.

What is the development plan for the next six months?
It will take at least four months, and based on the past year’s experience, most likely six months, to finish the scheduling and calendar part of Mallard, along with putting a polished user interface on what is already completed. The objective for the first six months is to have a program that is ready for beta testing. That will require setting up a server for beta testers to access publicly available information relevant to the information stored in their Mallard databases, and a server to handle synchronizing a Mallard database with Blackberry smartphones.

To get to that point with Mallard will require an additional $60,000 at minimum, and will probably need closer to $75,000. My wife and I have put $50,000 of our money into this program so far, so that I could work full-time on it for the past year. To continue working full-time on Mallard, I need to take home $4000 a month, which works out to a payroll cost of $6154 a month. I estimate that legal fees will be $6000 for the first six months, based on $2000 for a trademark search and registering a trademark for this program (which will not be called Mallard when it is finished, because that name has already been used for a computer program that claims it can “get your ducks in order”) and $4000 for incorporating a new company, and drawing up legal agreements for investors. Beta testers need to be able to synchronize their Blackberrys with Mallard and the most cost effective solution to provide over the air data synchronization with mobile devices (such as Blackberry smartphones) is a $10,000 “starter” package from a company called Funambol. I will also need to purchase existing software packages in order to import and export data with Microsoft Outlook, Outlook Express and Office and to produce reports (address listings, daily and monthly calendars) as PDF (Portable Document Format) files so that Mallard users can print and email their information. The services of a consultant will be required to set up a virtual private server, in order to provide online services for beta testers, and there will be monthly charges related to that server.

What is the development plan for the six months after that?
If the results of beta testing are positive, the next phase will involve spending money for the services of a graphic artist, a copy writer and a consultant to design and implement a professional looking website that can be used to sell Mallard licenses. There will be ongoing costs to host the website and operate the virtual servers needed to make Mallard work. $8000 will be budgeted for Google AdSense, so Mallard advertisements can be displayed along with search results, based on the keywords used for the search. Operating losses in the first six months that Mallard is for sale are probably inevitable, so a cash reserve will be needed to cover those losses.

Software bugs will need to fixed, and based on customer feedback, support for importing and exporting information with other programs and other mobile devices will be expanded. Development will start on a version of Mallard to be used in a corporate setting, starting with trials at a few professional offices. The objective is to start at least one of those trials by the end of the twelfth month. To take Mallard from the beta testing stage to the end of the twelfth month will require an additional $75,000, so I am looking for a total investment of $150,000 from outside sources; $75,000 now and $75,000 when Mallard is ready for beta testing.

What is your proposal for someone to invest in developing Mallard?
I am looking for 3-6 investors with $10,000 to $50,000 that they can put aside for up to 24 months before they get a return on their investment. Investing in a new software company is always risky, but splitting development into two phases reduces the overall risk. Once Mallard is in beta testing, it will be much clearer what the eventual success of Mallard will be, and based on that, investors can decide whether to continue with the second phase or not. I have limited expertise in tax accounting and corporate law, so professional advice will be needed before settling on the appropriate corporate structure but my thinking at this point is to offer 50% interest in the company to investors, with the sole proprietorship I've registered in Saskatchewan holding the other 50%. Investors for the first phase (getting Mallard ready for beta testing) would have the option to buy an equal amount of shares at the same price before those shares could be offered to anyone else for the second phase.

One advantage software companies have is the ability to expand without making large capital investments. The cost to produce more copies of a program or to add more users to a server is tiny. For Mallard the biggest cost will always be payroll, and once the company is on the way to profitability it should be easier to get bank financing to cover variations in cash flow that result from hiring new staff, or fluctuating sales. Therefore most, if not all, of the after tax income can be returned to investors.

Successful software companies can be extremely profitable. My projections show a before tax profit margin of around 65%, which translates into a monthly profit of more than $200,000 once sales reach 10,000 licenses a month. Based on the ownership structure I'm proposing, that would provide a 66% return on investment per month or **800%** per year (not including corporate income taxes)! There are no guarantees that this company will ever get to that level, but if it is any consolation, my wife and I still

have the most to lose should Mallard fail, and if Mallard ever takes off, we could hold our annual shareholders meeting in the Bahamas.

The Competition

Since 1983, hundreds of computer programs for managing personal information have been developed. I've been a manufacturer's sales rep for eighteen of the past nineteen years, have tried out a couple dozen programs of this type and have not found any program that saves time compared to using a paper journal like Day-Timer. I found a program called Info-Central that would let you refer to people and locations that you had previously inputted, by defining a relationship between every piece of information, but it still took too long to input your schedule or addresses and phone numbers, compared to pen and paper. Info-Central was discontinued in the mid-nineties, and I haven't found any programs since then that get any closer to what I want.

Today, the market for this type of program, commonly referred to as **PIM**s (Personal Information Managers) is dominated by IBM Lotus Notes in the "enterprise" market (typically large multinational corporations), Microsoft Outlook in the small to medium size business market, and free online calendars from Google and Yahoo in the personal market. **Lotus Notes** is a document management system that happens to have an email program, a calendar and a journal inside it, and most companies that use Notes spend millions every year to keep it operating. Lotus Notes has a well-deserved reputation as one of the worst computer programs ever written, but when it was released in the late eighties, it was light years ahead of anything else on the market, and companies such as General Motors spent millions to have custom applications written to work with Lotus Notes, mainly for sales force automation (which is what you call a computer program that keeps tabs on salespeople, especially salespeople who travel and have minimal supervision in their jobs). Shortly after I joined Michelin North America as a sales rep in 1997, we were given laptops preloaded with a custom program written to work with Lotus Notes, and I was required to report sales calls, keep mileage records and maintain a database of information about my customers with this program. Michelin still uses Lotus Notes for its North American salesforce, and it still takes two or three times as long to keep proper records than it did with the paper system it replaced.

IBM definitely makes more money from this software category than Microsoft does, mainly because Microsoft gives away its **Outlook** program with Office Professional. Like Lotus Notes, Outlook can be used in a corporate network, but it has far fewer features for distributing documents. Outlook's main feature is its email component, and the program Microsoft sells to manage Outlook users over a corporate network, called **Exchange**, is primarily an email server. Outlook's email component is much better than what Lotus Notes offers for email, and there may be millions of offices and independent professionals who keep every work related document in their Outlook mail folders. Nonetheless, Outlook was never really designed to handle gigabyte sized mail folders, and if you try to network more than two or three overstuffed Outlook databases, everything slows to a crawl. Outlook is probably the best looking PIM on the market, and it is possible to write add-in programs that customize Outlook to better suit the

requirements of corporate users, but that requires the services of professional programmers. It also requires the services of a professional to set up and maintain an Exchange server so that contacts and schedules can be shared between Outlook users. Therefore, most small businesses can't afford to connect individual Outlook users, even if their employees all have the program. Outlook was originally released in 1992 as a standalone program and can still be purchased separately from Microsoft Office for $110.

There is a category of software called **CRM** (Customer Relations Management) that many businesses use, and many of these programs incorporate some kind of calendar application, and are designed to maintain databases of information about customers, including addresses and phone numbers (for obvious reasons), so I'm including them in this market overview. The two most popular programs of this type, that are targeted at salespeople who tend to work on their own, are **ACT!** and **Goldmine**. ACT! is the most popular of the two and claims to have sold almost 3 million licenses over the past twelve years. For a comparison, IBM claims there are more than 145 million licensed users of Lotus Notes (a figure that is probably misleading since a number of Lotus Notes customers buy licenses in blocks of 10,000 even if they only need a few thousand) and the number of Outlook users (for email at least) is probably over 100 million. A single user license for ACT! costs $199.

IBM has had the SFA (Sales Force Automation) field to itself for a long time, which meant there was an opportunity for a competitor to step in and make bucketfuls of money. That competitor is **Salesforce.com**, which was smart enough to exploit IBM's biggest weakness, the need to use VPN's (Virtual Private Networks) to connect salespeople in the field with corporate servers. Salesforce.com sells a monthly service, not software licenses, and everything is done over the Internet. The technology behind Salesforce.com's web based applications is not very special, and there are a few Open Source Software clones, but Salesforce.com knows how to sell their services better than anyone else in this market. The big trend with large corporations has been to eliminate their in-house computer departments, and contract out everything to do with computers. Salesforce.com will handle tech support and provide customized services on their servers instead of the client's equipment, so the client no longer needs to have system administrators on staff to set everything up. This mostly trouble-free service comes with a big price tag (standard plan is $99 per user per **month**, and goes up quickly if you require any customization), but if you are a corporate sales manager with no desire to get involved with computers and networks, Salesforce.com can be very attractive.

Ten years ago, you could still find dozens of PIMs available in the $20 to $80 range, but nearly all of them are out of business now. Two things tended to kill them, one, it was difficult to modify their programs to enable synchronization with mobile devices such as the **Palm Pilot**, and two, some tried to incorporate email into their programs, and they couldn't compete with free email programs. That situation hasn't changed in ten years, in fact, synchronization with mobile devices such as **Blackberrys** and **iPhones** is far more important than it ever was to synchronize with Palm Pilots. The situation with email is even worse, web based email services offered by Yahoo and Google are starting to make inroads in the business world. Webmail isn't just for kids who don't want their parents to read their email.

Google and **Yahoo** also offer web based calendar programs, with the ability to synchronize your online calendar with your Blackberry, and it is possible to share your calendar online. However both programs offer very little in the way of customization, and there is very little you can do with the information you store on their servers. Both Google and Yahoo have hundreds of millions of users, but it is hard to gauge the popularity of their free web services, because more people sign up for free accounts than use the service on a regular basis. In May 2007, Yahoo announced that they had 8 million people use their Yahoo Calendar at least once in the month (apparently it was a record), which suggests that most people use online calendars on an infrequent basis.

Which brings up an interesting point. In 2005, 52 million **printed** calendars were sold in the United States alone. Worldwide, at least 800 million people use email on a frequent basis, often daily. It is possible that only one out of every twenty people who have access to computers use those computers to manage their schedules on a consistent basis. How many computer users only have email addresses in their computerized address books? There must be a reason why so many computer users chose to use pen and paper to manage their personal information (with the exception of email). The only reason I can come up with is that it is still easier to use pen and paper.

Mallard is designed to be a very different kind of PIM. There is no point in trying to tweak what is already available. Knowing what I wanted a PIM to do, I then geared my research to find techniques that could provide those capabilities. I could write a book about the considerations that went into every design choice, but that would be of little interest to anyone else. I do want to emphasize three guiding principles that separate Mallard from other PIMs.

1. Keep data entry to an absolute minimum, so you can actually save time by keeping your information in computer memory instead of on a piece of paper. If you have so much time on your hands you can keep everything neatly filed away (stored on paper, electronically, or in your head), you won't need this program. On the other hand, if you are using Microsoft Outlook, IBM Lotus Notes, Google Calendar or any other "organizer" program, you are undoubtably spending more time on time management than you would if you only used paper. Which is a shame, because there are a number of ways to manipulate your information more effectively with a computer than with pen and paper. After fighting with the PIMs already on the market, I decided to develop my own program.

2. Effectively manage your personal life and your work, separately when necessary and side by side when you need to coordinate one with the other. The line between work and home has become horribly blurred for ~~some~~ most of us. If only there was some way to put our homes (and family members) in suspended animation while we went to work, and never give a second thought to our jobs once we punched out at the end of the day. In real life, if you try to juggle commitments to more than one person or one organization you have keep your eye on everything at once and you have to make sure the commitments you are juggling never interfere with each other. The problem with every other PIM out there is that once you let others see your schedule or share your address book with a group (assuming the program even allows you to do

that), you lose control, and balls get dropped. You can always keep two (or more) sets of books (either electronically or on paper) to control who sees what, but if you don't have the time or inclination necessary to do everything twice, you have a problem.

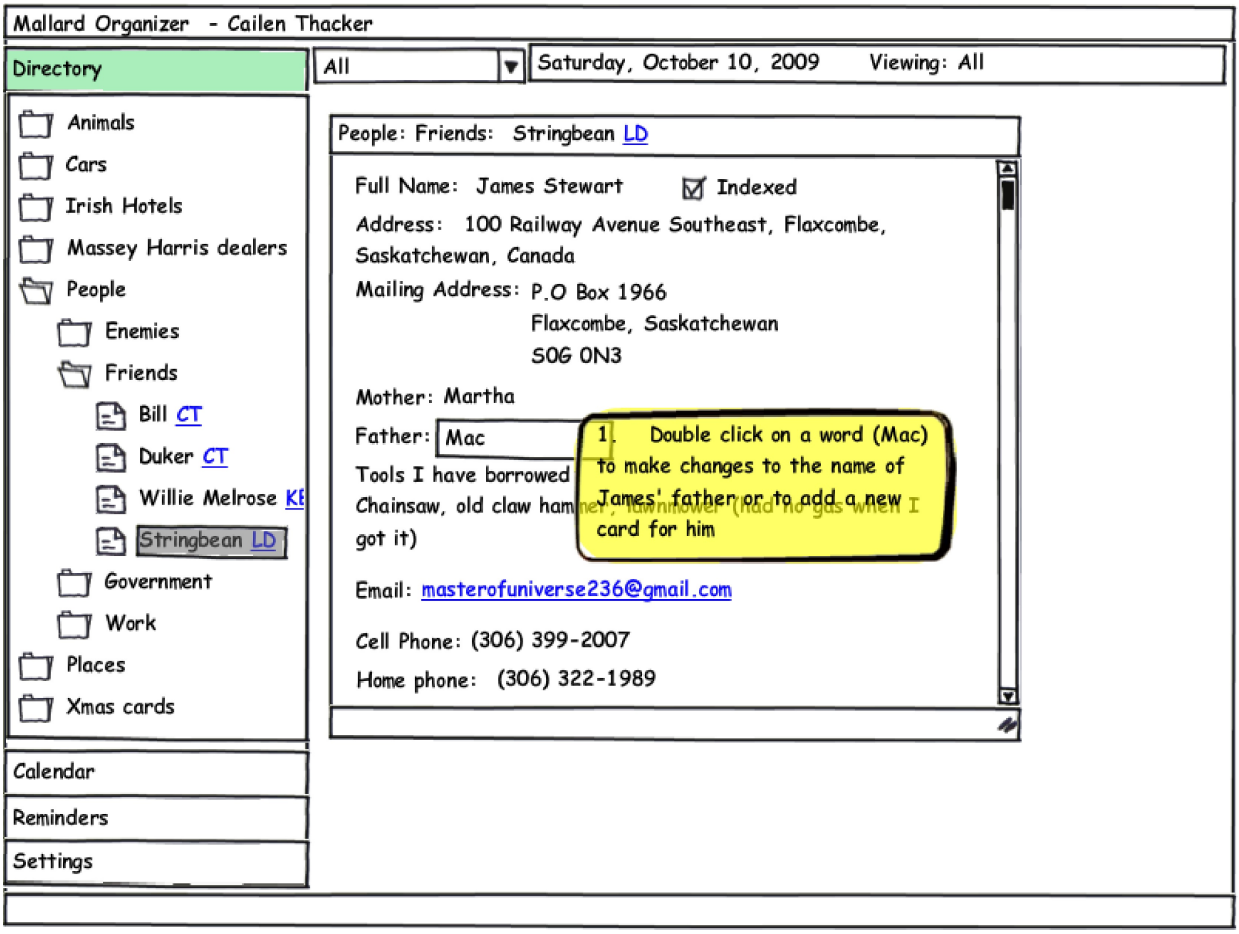
3. **Adapt to the structure and content of your information without extra work to customize it.** If you are a computer programmer, you should be able to either build your own PIM, with everything set up just the way you like it, or customize a program like Microsoft Outlook, so that it operates closer to the way you run your life. That would be good for you (assuming you are a reasonably competent programmer), but if you offered your personalized program to someone else to use, it simply wouldn't work as well for them. One thing I've discovered is that the information we use on a daily basis is fairly repetitive and consistent. We go to work at the same address time after time, make appointments with the same doctors and hairstylists, visit people in the same city or province as us, and so on. However, you don't call the same people I do, buy the same clothes as me, and probably could care less about the brands of tires on the trucks you pass on the freeway. We are all creatures of habit, but we don't all have the same habits.

Other PIMs use one of two different approaches: they are either so basic (one size fits all) you can't do very much with them, or they offer so many blanks to fill in (so if you know someone with five different phone numbers, you can enter them all) it takes longer to get through a form than it does to type in relevant information. Mallard will let you enter as much or as little information about anything as you want (or happen to have available to you) and you can have a different format for every person, business, or pumpkin pie recipe. What's more, if you want to change the way that information is laid out, you just do it with your mouse and keyboard. No programming ability required.

How Mallard Works

I've put together a mockup of how Mallard might look, to make it easier to explain **what** Mallard does. How it will actually **look** is still to be determined. The following illustrations and the notes to go with them are based on the database of a fictional user called Cailen Thacker.

1. There are two ways that you can look at the information in your Mallard database. One is called the Directory, and the other one is called the Calendar. The Directory holds all the information that isn't tied to a specific date or time. You can store information about people you know, people you work with, companies you do business with, reviews of Vancouver restaurants, and so on in the Directory. An outline of how that information is categorized will be displayed on the left, and a box designed to look like an index card will display the details on the right.



Every index card has a "title" at the top, which tells you where this piece of information fits in the outline, followed by the name you've chosen for that piece of information. A "piece of information" is going to be called an "object," even though "object" has a somewhat different meaning in computer programming. Usually, the information that we need to keep track of is composed of several separate pieces of information. For example, "Stringbean" is an object representing a person who I have categorized as a friend, and the Stringbean

object contains other objects, such as phone numbers, addresses and personal information, all of which are related to the person I call "Stringbean." Each object contained within the Stringbean object has a title, and those titles are what you see displayed on the card.

Objects in a Mallard database are based on a "parent" object, and new (or "child") objects inherit the information contained within their parent. All child objects in a Mallard database come from single parent families, but a single parent object can have an unlimited number of children. A child object can add to or remove some of the information that it inherited, but an object cannot change the type of information it represents. An object that refers to a physical location cannot be descended from an object that refers to a person. An object that represents my hair colour cannot be descended from an object that represents my '72 Maverick.

On the other hand, if you go back far enough in the "family tree" of the '72 Maverick object and a toaster object, they do have a common ancestor. That common ancestor is a generic object representing inanimate, mechanical things. The parent object of the Stringbean object is a generic Friends object, and Friends is descended from People. People is an object that all Mallard databases have, and the People object contains information such as Family Name, First Name, Date of Birth, and so on; which every object descended from People inherits.

If you don't know the birth dates for all of your friends (or any other category of People in your database), you can either leave that part of the card blank, or you can delete the part that reads: "Date of birth:" If you could care less about when your friends have birthdays, just find a Friends object without a Date of Birth object on it (because you previously deleted "Date of birth:") and make that object the parent of any new Friends objects.

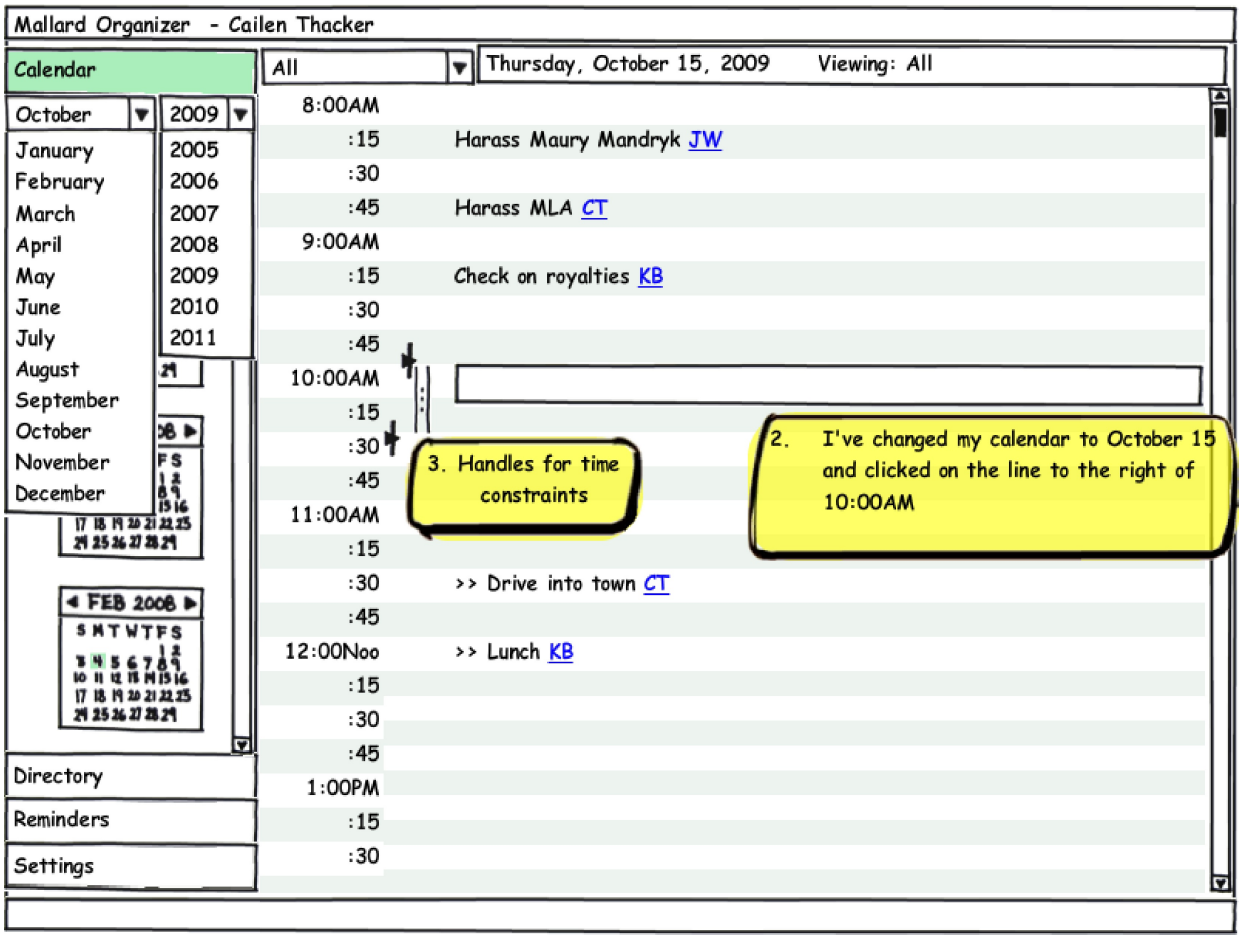
Making corrections to a card, or starting a new card is as simple as typing over one or more words on an existing card, or adding and deleting one or more words. You can also add more lines or blank spaces, if it makes it easier to read the card. There are no restrictions on where you place words on the card, and you can add as much or as little information as you want to a card. Mallard will check your changes against information already stored in your database, and ask you if you want make corrections to the existing card, or save the changed version as a new card.

The great thing about this is that you don't have to retype any information that doesn't change from the parent object to the child object. If you decide to save your changes as a new card, the card you made the changes to becomes the parent of the new card. If you simply want to add a new category, make the necessary changes in the title, and the outline for your information will be changed accordingly.

2. The other way to look at your information is with the Calendar. You can assign a date and time to information that involves you (the owner of the database) doing something with someone or something (which we will call an Event), and Mallard will show you all the Events assigned to similar times. The left side of the screen provides a quick way to get to a specific day, week or month, and the right side displays the information connected to the period of time you selected on the left side. The time scale on the right side can be adjusted as needed, so you could see an entire month on the screen at a time, or you could see a single hour, divided into minutes and seconds. Most people will never need that level of fine detail, but if you do, Mallard can handle it.

I've scribbled a note to meet Kerri at Grainfields on 8th Street at 10:00 Thursday, along with

her phone number 374-1999. I open up Mallard on my computer, click on October 15th in the monthly calendar on the left side of the screen and then click on the line across from 10:00 AM on the daily agenda on the right hand side. The line I've clicked on turns into a long, narrow box with handles attached to the left edge. When I opened the program, the daily agenda part was open to today's date, but it changed to the 15th when I clicked on that date in the small monthly calendar. So far, you could do the same thing with every other PIM out there, with the same amount of mouse action.

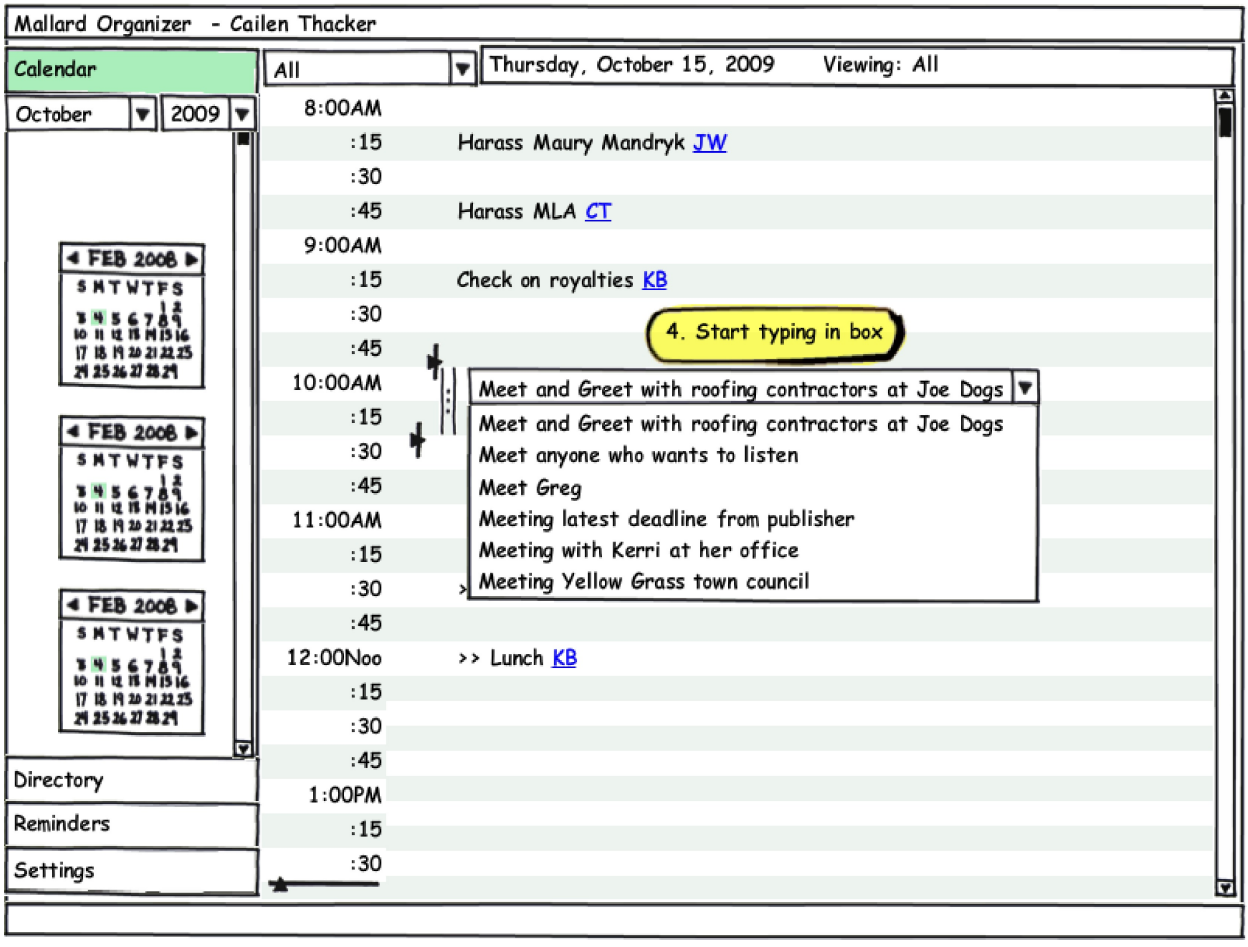


3. The two little handles on the left side of the long, narrow box are time range indicators for the scheduled starting and ending time of an Event (which is the meeting with Kerri, in this case). Each handle is long enough to cover 10 minutes on the time scale on the left side of the daily agenda, and the markers in the centre of the handles are set at 10:00AM and 10:30AM. The default length of an Event for me is thirty minutes, so if I add something to my Calendar, the markers will be set thirty minutes apart until I move them. Most of my Events don't start or end at exactly the scheduled time, and when I look at my Calendar I would like to have some idea of how much of the day could be taken up by this meeting. By setting a time range for the meeting to start and end, I can plainly see if I am bunching up scheduled Events. I've set my Mallard program to automatically block time five minutes before and five minutes after the scheduled time, but if I need to change those time constraints, I can drag either end of the time range handle. What if I want to change the length of the meeting? I can either move the markers, or I could double-click the Event and change the default length for just this Event. The advantage of manually changing the length of an Event is that every time I add that particular Event to my Calendar, it will automatically use the length of time I've set it to, without changing every other new Event.

If I was a doctor, my defaults might be 6 minutes for the length of the Event (seeing my patient), and the span for the start time would go from the exact appointment time to 6 minutes before lunch or 6 minutes before my clinic closes, depending on whether the appointment is before or after lunch. The end time span would be adjusted accordingly. These limits would apply to all my appointments for the day. Why? I have a history of getting behind in my appointments, and my patients never show up early so I never see the patient before the scheduled time. My clinic is closed for lunch and when my assistant goes home, we close the clinic. Therefore an appointment for the morning could get pushed until six minutes before lunch before it has to be rescheduled, and something similar applies to afternoon appointments. Giving every appointment these time constraints means that Mallard can calculate if I'm getting overbooked (based on who I've already seen), even though I seldom see a patient at the actual appointment time.

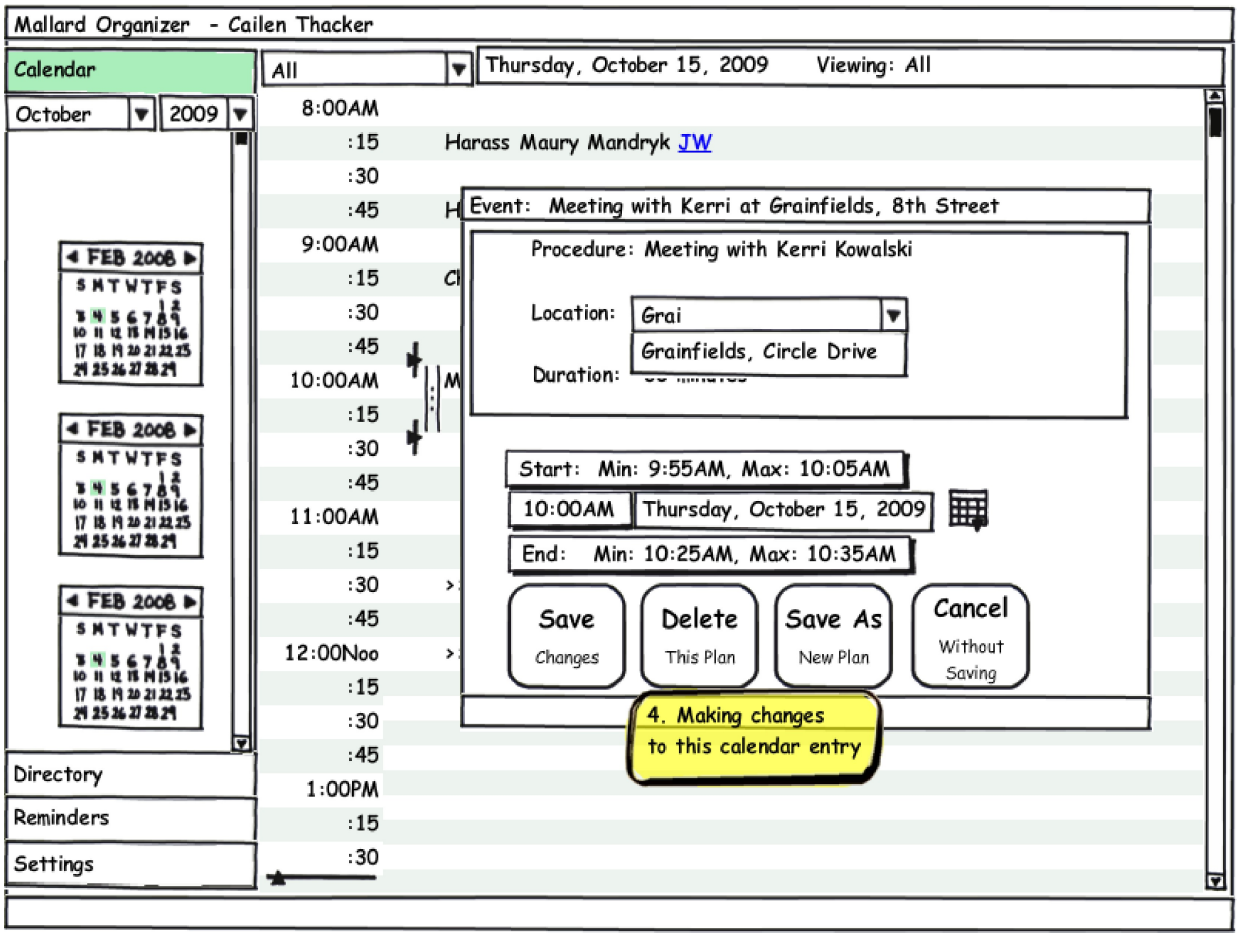
A good office assistant can do the same without doing the mathematics that Mallard does, but there is no other PIM out there that can adapt to real life **and** provide useful information in the same way. I could buy custom software to handle scheduling for my office, but it wouldn't be much help managing my family time, or my tee times (and I don't want my wife or my golf buddies to see my office schedule). And if I needed to change the defaults for that custom software, I would have to pay a computer programmer make the changes. Not only is that expensive, but my assistant is not going to appreciate having to translate computer jargon in order to explain to the programmer what needs to be done.

4. I click in the long, narrow box and start typing. Once I have typed in 4 letters ("meet") and pause for half a second, something neat happens. The box fills up with the first suitable



match from all the Events I've already entered. In this case, I get "Meet and Greet with roofing contractors at Joe Dogs", which isn't what I want, so I click the triangle at the end of the box, and get a drop down list with every Event that starts with "meet", "Meet", "MEET", "MeET" and so on. I keep typing " ing with Kerri", pause for half a second again, and this time I get "Meeting with Kerri at her office", because three months ago I went downtown to talk to her about mutual funds. This is as close as I'm going to get without entering some new information, so I double click in the box.

A new window pops up that looks like an index card, and it has all the information related to that meeting three months ago. "Meeting with Kerri at her office" is on the top line and I change that to "Meeting with Kerri at Grainfields, 8th Street". Below that line is a line that reads: "Procedure: Meeting with Kerri Kowalski". That information still applies, so I look at the next line which reads: "Location: Suite 1320, PCS Tower", which needs to be changed to "Location: Grainfields, 8th Street & Grosvenor", so I double click somewhere between "Suite" and "Tower", and another drop-down box appears. I start typing "grai", pause and the box fills up with "Grainfields, Circle Drive". I keep typing until "8th Street & Grosvenor" is in the box, then I hit the TAB key, and the drop-down box disappears. Along the bottom of the card is the appointment time and the limits for the meeting to start and end, and I decide that I don't need to change any of those values, so I click on a button labeled "Save As New Plan?" The window that looks like an index card disappears and I'm back at my calendar.



What just happened? All Events (as far as Mallard is concerned) have three parts: a Procedure, a Location, and a time Duration. A Procedure is an information object that describes what you are going to do. Mallard comes preloaded with dozens of generic

Procedures, such as "Grocery shopping", "Doctor's appointment", "Parent-Teacher interviews" and so on, but what happens if you have something that isn't already in the database, such as "Rescue Taliban hostage"? You just type that in beside "Procedure:" and the next time you need to schedule another hostage rescue, it will be there waiting for you. And if you decide you would rather call it a "Task" instead of a "Procedure", just type over "Procedure". It is just that easy

The whole idea behind connecting a Procedure object to an Event object is to reuse information that you have already entered into your database (or information that was preloaded into your database when you bought your Mallard license). If I don't need any more information about what happens when I have a "Meeting with Kerri Kowalski" I don't need to type anything, but if I want to, I can break down a Procedure into as many intermediate steps as necessary. As long as all of these intermediate steps are connected to the meeting, and there isn't a break between the first step and the final step, these steps can be part of the Procedure object. For instance, a "Fly from Saskatoon to Toronto" Procedure could be subdivided into "Call a cab," "Ride to airport," "Pay cab fare," "Check in luggage," "Get boarding pass," "Go through security," "Board flight AC8871," "Deplane," "Pick up luggage," "Take a cab to the Constellation Hotel," and "Hand my wallet to the cabdriver."

Why bother? Because Mallard is programmed to know that you can't be in two places at the same time, and some steps are contradictory and can't be performed at the same time, and you can set limits on which steps come before other steps, and which have to occur at the same time. Most people will never need this level of detail, but if you do, Mallard is the only PIM that can manage those details. No program can figure out the complex relationships between your actions without human input, but Mallard can at least work with what you do input and present that information back to you in a form that makes sense.

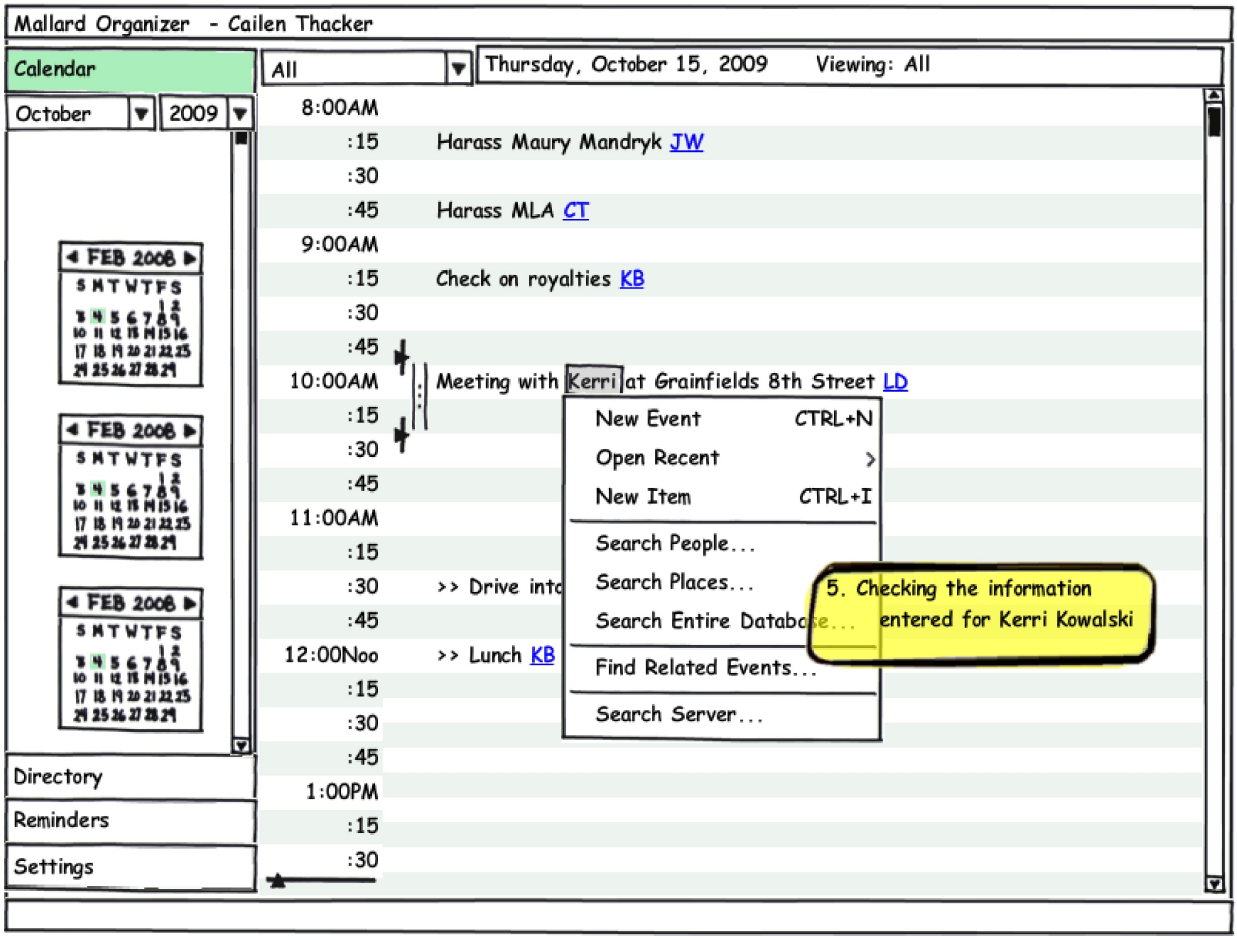
You still can't enter an Event in your Calendar without giving it a Location. There is no point in scheduling an Event if you don't know where it is going to take place. Mallard will try to match what you enter for a location with a Location object that already exists in your database, but in my example the program had to automatically create a new Location object with the title "Grainfields, 8th & Grosvenor." This new Location object is based on the object titled "Grainfields, Circle Drive." The new object inherits the information that Grainfields is located in Saskatoon, and if I have added information to the "Grainfields, Circle Drive" object about the type of restaurant it is and so on, "Grainfields, 8th & Grosvenor" will have the same information.

Almost everything we do more than once or twice takes about the same amount of time each time we do it. Continuing with the principle of reusing information, every Event has a time duration associated with it, and every time you put that Event in your Calendar, the same amount of time will be assigned to it. Because Mallard works with words, you could enter "half an hour" or "30 minutes" and Mallard will try to match up what you enter with a time object in your database, and convert what you entered into seconds. Why seconds? Because your computer can't consistently perform two separate tasks (such as displaying a warning and playing a song) with any more accuracy than a single second between them. The current version of Mallard has a limit of 68 years and 18 days for a time duration, but future versions could extend that to 292 billion years, although doing so would require your computer to work harder and make your database file larger.

The big benefit of having the time duration stored with the Event object is that when you add an Event to your Calendar you don't have to manually change the time it takes. Even

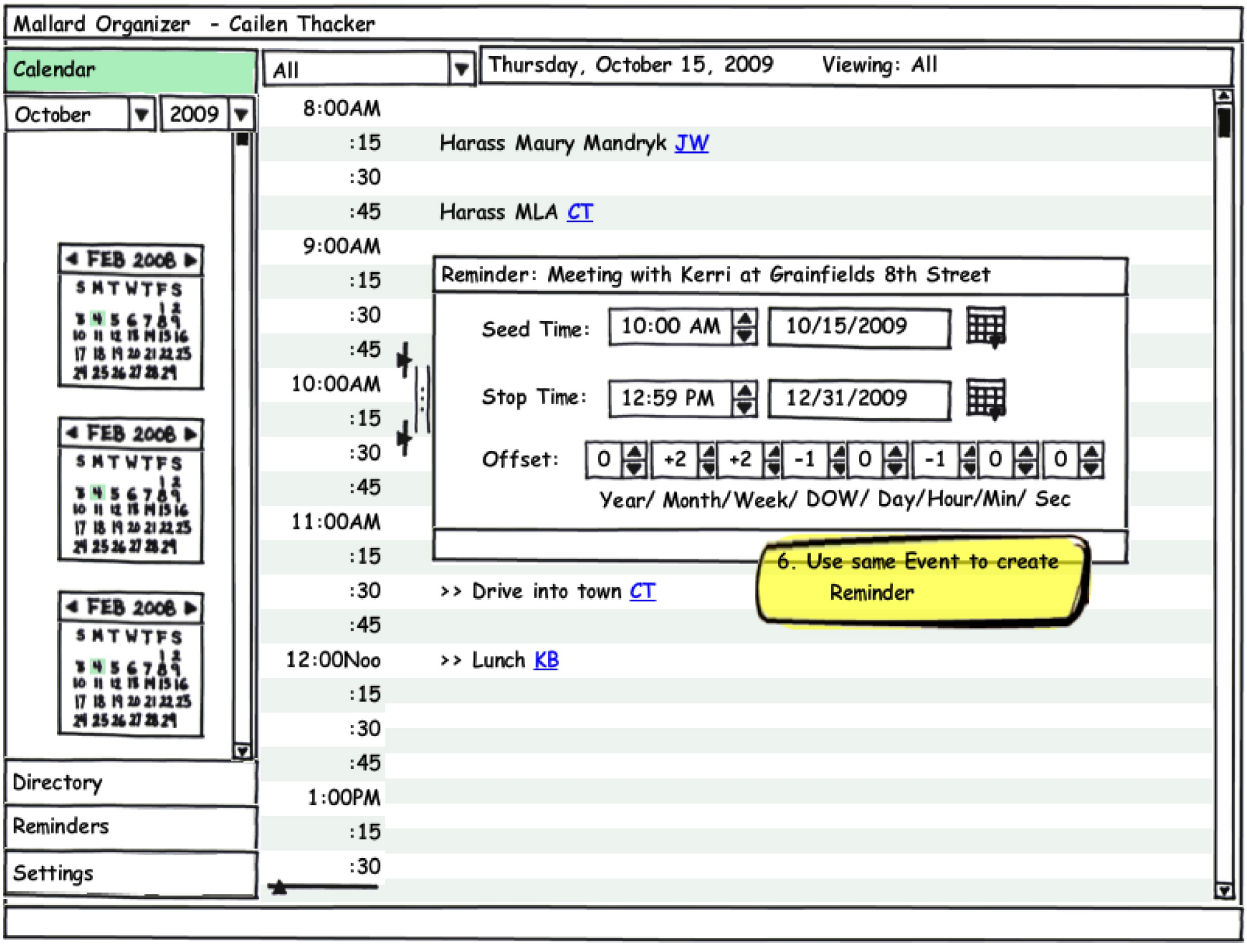
the simplest PIM lets you change the default length of your appointments, but if your appointments are different from each other, the default time duration will be wrong most of the time. If I discover that my meetings with Kerri are tending to last 45 minutes instead of half an hour, I change the Duration accordingly, and the next time I schedule a meeting with her, three-quarters of an hour will be assigned, and that will continue to be the case until I find it worthwhile to change the Duration again. If the same Event takes half an hour sometimes and an hour other times, I can store two Events in my database with the only difference between them being their different time durations. The extra mouse click to pick one Event over the other is less work than manually changing the time duration.

5. What about Kerri's phone number? I could have checked if I already had it in my database when I was making changes to the Event, but now I'll use my mouse to select her name in the line on the calendar that reads "Meeting with Kerri at Grainfields, 8th Street" and right-click. I get a short menu that offers to "Search people...", "Search places...", "Search entire database...". I get a list of names that start with "Kerri" (which is a list of one), click on "Edit information" and up comes another window that looks like an index card. This card has Kerri's office address, a work number and a cell number and a link to Manulife's website. I don't need to enter the phone number I had, but I remember that she said she doesn't work Thursday afternoons, so I add a line to the bottom of the index card, and type that in. If I want to change the layout of the information on this card, I can cut and paste to my heart's content, add and delete lines, add and delete categories, basically whatever is



possible with a keyboard and mouse. Mallard doesn't care what it looks like but it keeps track of which words on the card are related to other pieces of information, and will ask for your input whenever it isn't clear what should be done with the changes.

6. If you click on ">> Drive into town" or ">> Lunch", you won't see any time constraint handles. Those two entries are Reminders, the meeting with Kerri is a Plan, and while both Reminders and Plans are based on Events stored in your own database, and have start and end times assigned to them, they have different functions. Plans are Events that will occur or have occurred at a single point in time, and if you want to repeat a Plan five times, five separate Plans will be stored in your database. Reminders are used to make Events show up on your Calendar, without keeping a record in your database of when the Event actually occurs. The Reminder for Lunch at noon is a good example. I don't want to schedule appointments during lunch, but I don't want to record all of my meals in my database, either.



You can always convert a Reminder into a Plan, but Mallard will ignore Reminders when calculating if there is enough time in the day or week (or month, or year) to complete all of your Plans. Therefore, there is no need to give time constraints for Reminders, and you can add and delete Reminders without any effect on your Plans. If you skip a Reminder it is not a big deal. You can also use Reminders as alarm clocks. You can set a Reminder to trigger a program built into Mallard that can display a warning, play a sound or even send an email with a specific message. The reason why you can't set an alarm for a Plan, is that you don't want to save a record of every time an alarm goes off in your database. Alarms are there to "remind" you of something you should do, but triggering alarms is not something that you want to keep a permanent record of.

Virtually every PIM has a function to repeat appointments every day, or week or month, but Reminders can be set to repeat themselves in ways that can't be accomplished with other

programs. Mallard does this with something called offsets. To create a Reminder to meet Kerri during the second week of December, but on Wednesday at 9:00AM instead of Thursday at 10:00AM, I would use the same Event (so the Procedure, Location and time Duration do not change), and set Seed Time to the same time as the Plan I have entered for October 15th. Seed Time is the first date and time that a Reminder occurs. I don't expect to keep meeting her after the end of this year, so Stop Time is set for December 31st, 2009. Stop Time is the date and time when the Reminder stops repeating. The Year offset remains at 0, Month offset is +2 (December is two months after October), Week offset is +2 (I want the second week of the month), Day of Week offset is -1 (Wednesday is one day ahead of Thursday in the week), Hour offset is -1 (the next meeting is an hour earlier), and the Minute and Second offsets remain at 0.

If I change my calender to December 9th, there should be ">> Meeting with Kerri at Grainfields, 8th Street" beside 9:00AM. The Stop Time can be earlier than the Seed Time (so the Reminder will repeat itself going backwards in time), and the Seed Time can be any time in the future or past. If you set Seed Time to a future date, it is possible to set Reminders to occur at times relative to public holidays that occur at irregular intervals, such as Easter, Chinese New Year, Islamic religious festivals, and so on. If you want a Reminder to only occur once, make the Stop Time the same as the Seed Time. Why would you create a Reminder that only occurs once? Because you want it to show up on your Calendar when you do your planning, but you don't want a permanent record of it.

7. There are times when you will want to look at your Reminders in a list, instead of on your Calendar. That is what the Reminders tab on the left side of the screen is for. If you have several Reminders set up, it is easy to forget when they will next occur (that is why you set them up in the first place, to be a reminder to you). It may be several days or weeks (or months) before a Reminder repeats, so it is easier to go directly to a list of Reminders than to scroll through the Calendar until you find it.

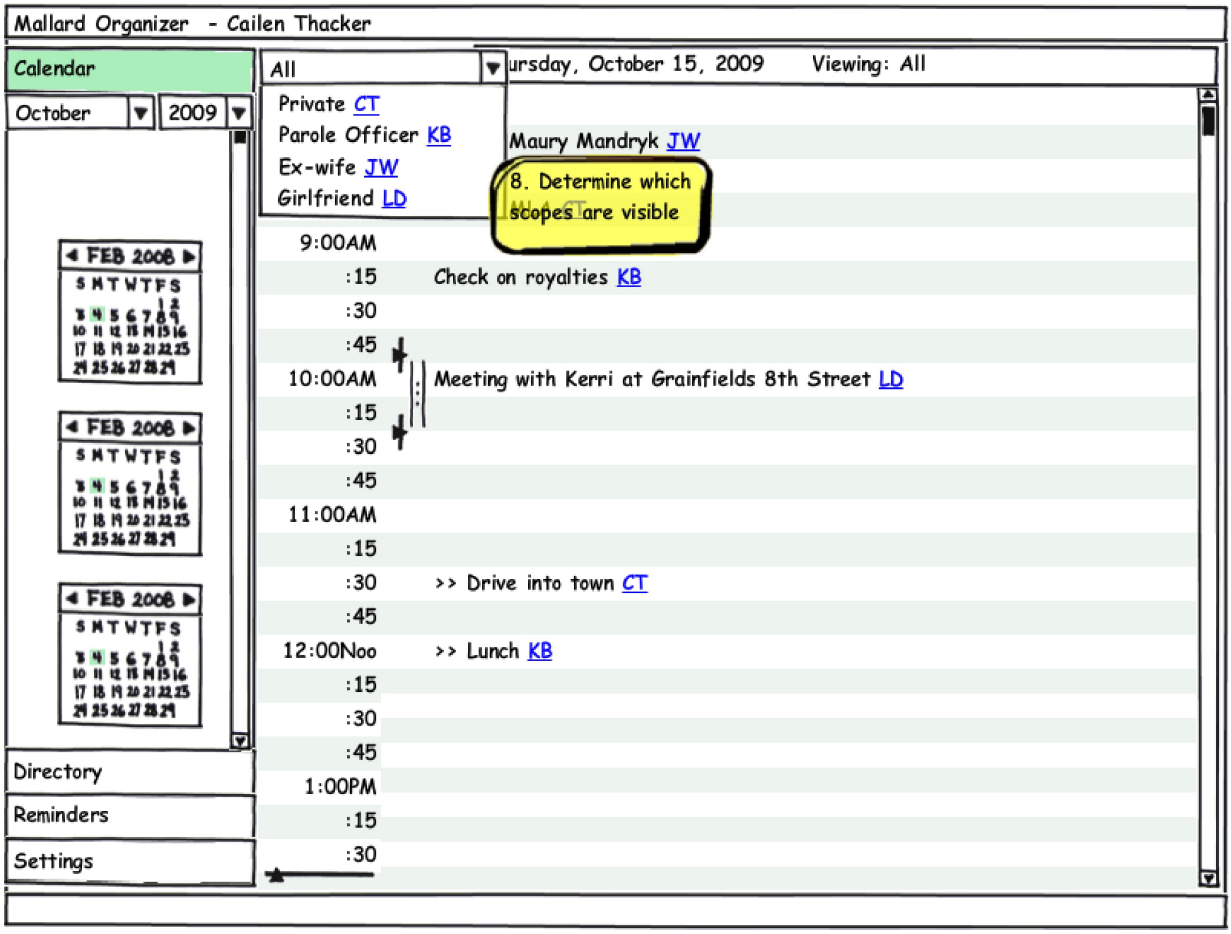
   You will not find a tab for To Do lists in Mallard. The main reason other PIMs treat your to do list as something separate from your schedule is that they don't have time constraints. With Mallard, time constraints allow you to set very loose deadlines for all the things you need to get done. If I can meet with Kerri at any time, as long as it is in the month of October, I set the start time constraint for that Plan to between one second after midnight on October 1st and midnight October 31st. The end time constraint is the same. Then, whenever my Calendar is open to some day in October there will be a line at the top of the Calendar without a time beside it, and the number of Plans that **could** occur during the time displayed on my screen will be shown on that top line.

   Clicking on that line will open a drop-down box with those Plans, and if I select one of them, I can either drag it to a specific time on the Calendar, or I can look at the card for that Plan, and decide if I want to add it to the time period displayed on the Calendar. Once it gets late enough in October that the Plan **has** to occur within the time period displayed, Mallard will automatically display it in the Calendar, and shorten the time constraints accordingly.

   There is no point in having a to do list if the items on your list don't have deadlines. In my world, if I don't assign a deadline to a task, it will never get done. Once I give a deadline to a task, I can save it as a Plan or a Reminder, and then I can see it in my Calendar, and incorporate it into my planning. In real life, we only need to worry about two kinds of tasks; tasks we **should** do, and tasks we **must** do. Every thing that **must** get done, also needs to be completed by a certain time. If you try to keep putting off your **must** get done tasks,

someone, somewhere will force you to put aside your **should** get done tasks and get started on your **must** get done tasks. Therefore, if you really want to do one of your **should** get done tasks, you need to upgrade it to a **must** get done task. That's human nature, but Mallard is the only general-purpose PIM with flexible time constraints so you can keep your to do list in front of you in your Calendar.

Mallard cannot tell you in what order you should do your Plans. You have to make that decision yourself. There are computer algorithms that can suggest which decisions will produce the best results, but using a computer program to do that requires the input of exact values for a large number of assumptions, and even then the program user will typically have to decide which algorithm to use for each specific situation. Unless you have a degree in Statistical Analysis, you probably aren't qualified to make that decision. You are qualified, however, to make your own decisions on what to do with your own time. In some situations, Mallard will be able to tell you if your decisions are feasible or not (in other words, either possible or impossible), but that will depend on the amount of detail you provide regarding the Events in your Plans.



8. Did you notice that Plans and Reminders (and the objects in your Directory) have two underlined letters at the end of the title? Those letters refer to the scope of the object, which is how you control the information you share with others. Every Mallard database has two scopes built into it, Public and Private, but you can set up your own custom scopes, and control access to the information in those scopes. There is no technical limit to the number of scopes you set up, but too many extra scopes will add to the complexity of your database. In this database, "CT" is shorthand for "Cailen Thacker", and is the label I've given to the Private scope. "KB" is shorthand for the scope that contains all of the information related

to my parole officer, "JW" is shorthand for the scope related to my ex-wife, and so on.

Information in these scopes can only be related to information from the same scope, or the Public scope. For instance, when I click on the Event object for the meeting with Kerri, the drop-down box will only show me Events from the LD or Girlfriend scope, as well as any Public Events. Typically, scopes will be set up for work, home and perhaps a club or society we belong to. Since it is our database, we can view all of the scopes at the same time if we want, or if we want to concentrate on just our work schedule (or company address book), we can select just a single scope to look at, or whatever combination we want. Any calculations done by Mallard, such as determining if we have run out of time to complete all of our Plans, are done with the scopes that are visible. Regardless of how many scopes are visible, each object can only be comprised of Public objects or other objects from the same scope. You can copy an object from one scope to another, and Mallard will add the necessary related objects to the other scope. Some duplication of information in your database is inevitable, but you only have to input it once. This duplication might be a problem if you have tens of thousands of addresses taking up space in three or four scopes, but in our own databases that is highly unlikely.

Information related to physical locations (such as City, Province, Country, area code, time zone, etc.) will be placed in the Public scope as much as possible. Your Mallard database will be able to import Public information from servers over the Internet, and when you purchase a license for a Mallard database, it will be preloaded with information suitable for the personal profile you have selected. While connected to the Internet you can search those servers for information related to a word or phrase, and those Public Mallard servers will provide access to far more information than you could ever input yourself, potentially everything from postal codes to train schedules.

The primary reason Mallard organizes your information by scope, is to allow you control over the information you share with others. Other PIMs let you designate contacts and tasks as private, but what if you are a consultant and you want to share information with more than one client? You set up two different scopes, one for Client A and one for Client B. Importing and exporting information with mobile devices, such as Blackberrys and iPhones, or shared online calendars is still possible, and you can use scopes to control who gets to see what information in those situations as well.

Additional features planned for Mallard

- *Extending the Windows Clipboard so that whenever text is selected in any program, it can be pasted directly into your Mallard database.*

  Whether it is a web page, email, or Word document that you are reading, the right-click menu can be extended to provide an option to paste selected text into a Mallard database. Since Mallard tries to match words and phrases in new text to information already in the database, Mallard will automatically relate the new text to existing information as much as possible, reducing the amount of typing the user needs to do.

- *Check web pages for updated information without opening a web browser.*

  What happens if a website has information you need, and that information changes from time to time? Mallard can retrieve the appropriate web pages, search for the specific information you need, and update your database if that

information has changed, all without opening a web browser. If the hours of operation for the city dump change from time to time, and the city puts those changes on their website, Mallard could retrieve that information for you every time you schedule a major cleanup.

- *Import and export information formatted according to the vCard and iCalendar standards.*

  Not only do most of the mobile devices accept Contact information in the vCard format and scheduling information in the iCalendar format, but most email programs, and web-based calendars do as well, so address books for your email program can be kept up to date with your Mallard database, and online calendars can be used to "publish" Plans from your Mallard database, although features like time constraints cannot be displayed with these online calendars. This could be used to make schedules for sports tournaments and other public events visible to the general public on websites. Mallard can be used to do the planning and once the schedule is set, it can be uploaded to a website.

- *Store references to computer files in a Mallard database.*

  It is not possible to store images or anything other than text (which includes numeric values) in a Mallard database, but file names and their location on the computer can be stored in the database. By double clicking on the filename while it is displayed in Mallard, the program needed to view and edit the file can be automatically launched with the file loaded. File references can include digital photos, spreadsheets, music files, any file that has its type registered on that computer. Mallard will never replace iTunes or your favorite photo gallery program, but you could use this feature to keep track of computer files related to a specific person or company, and open those files without have to go to Windows Explorer.

Sharing information between Mallard databases

- *Users can only add or delete information in the Public scope, not modify or create new information in this scope.*

  This is actually a good feature, and provides a basic level of consistency in all Mallard databases. Since Mallard can search for Public information on Mallard servers, the server can make various authoritative databases available to Mallard users. Consider this example: it costs $3750 per year to get the Canadian postal code database from Canada Post, and if there are 100,000 Mallard users in Canada, that works out to less than $0.04 per user per year. If you are adding an address to your database based on an existing address in the same city, you already have the city and province displayed, so once you enter the house number and street name, you can have Mallard search for the right postal code, and a few seconds later, it will be automatically inserted in your database. If Mallard becomes popular enough, various organizations and businesses will want to make their information freely available to Mallard users, in some cases even pay for it to be hosted on a Mallard server. Compare this to having to open your web browser, search for the web site you want, find the right web site from the search results, follow the links to the information you want, manually enter

any information requested by the web site, and then manually copy the information provided by the web site back into the program you are using.

- *Employers can maintain consistency of information by putting shared information in the Public scope on their own servers, and controlling who can access those servers.*

  The key difference between the Public scope and other shared scopes in a corporate setting, is that users (employees, for instance) cannot make changes to Public information, but they can make changes to information in other scopes and have those changes sent to the company's server. Some information (like customer ID numbers) should be protected from accidental changes, but it may be desirable to give employees the ability to submit changes to a customer's mailing address, or to update contact information. The server will handle any conflicts between information that has been submitted.

- *A Mallard server can be hosted by one company on behalf of another company.*

  This is stealing a page from Salesforce.com. Small business that cannot afford to hire staff to run their own server, but need a way to share schedules and client information between employees, even when they are out of the office; would pay a monthly fee to the company that sells Mallard licenses (i.e. you and I) to have their own shared Mallard database available anywhere there is an Internet connection.

- *A Mallard server can have thousands of scopes contained within a single database.*

  Mallard servers do not need to display the contents of a database, just manage the traffic between the shared database and individual databases. Each employee can have his or her own scope on the corporate server, separate scopes can be set up for offices or departments, and the electronic distribution of documents can be handled by putting references to the appropriate files in appropriate scopes. Every time a Mallard user attempts to access a Mallard server, a unique ID from the user's database is sent to the server, to authenticate the user making the request. The information transmitted is encrypted, and is meaningless outside of a valid Mallard database. Not only does this prevent hackers from getting into the company database, but if the database is messed up, intentionally or otherwise, by someone who has been granted access, that person can be identified and dealt with (legally, of course).

This is just a selection of what Mallard can do with your personal information (technically, any information relevant to anything you do in your lifetime). I wanted a program that would be useful for anyone that needs to do a better job of organizing their life, whether it is to keep track of monthly trips into town to get supplies for their wilderness cabin, or to keep up with three jobs, two kids, one spouse and duties as President of the local Caribou Lodge (there aren't any computer programs that can help someone crazy enough to juggle two spouses). Mallard is not designed to manage large amounts of information at once, it is designed to handle a wide range of information for one person at a time.

For the most part, electronic information exchanged between human beings is handled using human techniques. There aren't any email programs that can read your emails and make sense out of them. Google Search can find a word or phrase in trillions of web pages, and do an outstanding job of calculating the most relevant results based on patterns in the words on those pages, the popularity of the web site, and several other factors (mostly human factors, by the way), but all of Google's processing power cannot tell you if the information contained in those pages is correct. Mallard is not a particularly intelligent computer program; it is designed to take whatever information we put into it, keep track of that information, and display it in a format that helps us to decide what we want to do with that information. Mallard cannot determine what information should be exchanged with other people, it simply facilitates that exchange, should we decide that is what we want to do. It is a relatively simple tool, but it does some things that no other single computer program on the market today can do.

I have no illusions that I have special abilities that other computer programmers do not have, but I might be the only traveling salesman who disliked those other programs so much he quit his job and did the research necessary to figure out how to design a better PIM. I have asked other people what they like and don't like (and the don't likes outnumber the do likes by a wide margin) in the programs they use to manage their personal information, and I have eleven years experience as an unhappy user of the super-heavyweight in this market, Lotus Notes.

I've put additional information about how Mallard works from a technical perspective in a separate section at the end of this document, to give you an idea of what I got out of that research. I truly believe that approaching the design of Mallard from the perspective of how a PIM should work, instead of trying to find ways to improve what was already available, can result in a program that eventually convinces millions of people to use a computer instead of pen and paper to organize their life. Investing in Mallard is still risky, but showing what this program can do, should help you gauge how likely it is that Mallard will be a success in the marketplace.

Financial Notes

Fixed Costs:

- do not change with number of units sold or produced

1. Ongoing monthly expenses

For the first six months:

Payroll		
My salary	5420.00	
Vacation pay	325.20	
Employer's CPP contribution	269.95	
Employer's EI premium	139.15	
Subtotal:	$6,154.30	
Cellular (minimum voice and data plan for Blackberry and associated fees)	$50.00	
TOTAL	$6,204.30	per month

Once Mallard is in beta testing:

Payroll	$6,154.30	
Cellular (minimum voice and data plan for Blackberry)	$50.00	
VPS (Virtual Private Server) hosting	$120.00	
FogBugz (hosted bug-tracking service)	$30.00	
TOTAL	$6,354.30	per month

For six months after Mallard is released:

Payroll	$6,154.30	
Cellular (minimum voice and data plan for Blackberry)	$50.00	
Server charges (could be a combination of VPS' and Amazon's S3 service)	$300.00	
FogBugz (hosted bug-tracking service)	$30.00	
AdSense advertising campaign	$1,500.00	
Mailbox service	$40.00	
TOTAL	$8,074.30	per month

Once sales exceed 1000 licenses a month, this company will need to hire extra employees or contract out most software development, or more likely, do a combination of the two. My estimate is that it will take six months to reach the breakeven point in sales (which is 400-500 licenses per month), and once Mallard is profitable, further investment may not be necessary, and would only happen with the approval of existing shareholders. For that reason, these financial projections do not go beyond six months after Mallard is released for sale.

Even if it takes longer than six months to reach breakeven point, the only expenses that might increase in the meantime would be advertising and promotion. The appropriate level of online advertising can usually be determined within two or three months, so increased spending can be tied to increased sales. Whatever the most cost effective level of advertising is (and it could be $0), it should not change the amount of investment needed. If advertising exceeds $1500 per month, that extra cost will be offset by increased sales.

2. One-time expenditures

Legal:	
Trademark search and registration	$2,000.00
Incorporation and shareholder agreements	$4,000.00
End User License Agreement (EULA)	$2,000.00
Software for program development:	
Components for Microsoft Office COM automation	$700.00
Components for creating PDF files	$350.00
Empower ISV subscription	$575.00
InstallAware developer license	$1,000.00
Internet security certificates (annual expenditures)	
SSL certificate (for secure transactions)	$75.00
SSL certificate (for Mallard server)	$75.00
Trust certificate (to avoid security warning when installing Mallard in Windows Vista and Windows 7)	$375.00
Internet domain registration (annual expenditures)	$500.00
Graphic design (for the program itself and for the web site)	$5,000.00
Copy writer (for web site and press releases)	$2,500.00
Consulting:	
Funambol Starter package (to set up	$10,000.00

server for over the air synchronization with mobile devices)		
Set up VPS for Public Mallard server (SOAP interface and backend databases)	$10,000.00	(uneducated guess at this point)
Set up server to manage licensing and backups to Amazon S3	$7,500.00	(also an uneducated guess)
Design of web site for promoting and selling Mallard licenses (same site provides customer support)	$10,000.00	(the quality of the web site determines what this amount is)

If it takes six months to get Mallard ready for beta testing, two months of beta testing before it is ready to be released for sale, and six months of sales to reach breakeven point, total fixed costs will be just over $147,000. The amount spent on consulting can be reduced, but only at the expense of my time, which means that it will take longer to get Mallard ready for sale.

In the first six months the following one-time expenditures will be incurred (and paid with the first phase of investment):

- Incorporation and shareholder agreements
- Trademark search and registration
- Software for development
- Funambol Starter package
- Consulting to set up VPS

The total of these one-time costs and six months of ongoing costs is $66,000. When Mallard is ready for beta testing, the shareholders can decide if it is worthwhile to continue with the second phase of investment in order to bring Mallard to market.

Variable Costs

-are associated with each unit that is produced or sold

1. Payment processing

Fastspring is a highly rated, full service payment processor similar to PayPal that charges 5.9% plus $0.95 US per transaction (or a straight 8.9%). This is more expensive than setting up your own merchant accounts for credit cards, but there are no set up charges with Fastspring, and the most it will cost to process a $50 transaction is $3.90. Becoming profitable will depend more on growing sales than cutting costs for payment processing, and Fastspring can accept payment in 17 currencies and all major credit cards.

2. Customer support

It is estimated that it costs $30 every time customer support is delivered over a telephone line, but no one's life should be at risk if they cannot open their Mallard database, so most customer support can be handled through email. The customer support department (which will be myself in

the beginning) can process three or four emails in the time it takes handle a telephone support call, which will reduce the cost of customer support. The web site can also be an important source of information on how to use Mallard, and how to handle frequently encountered problems. The amount of customer support required is higher with new products (because more bugs are waiting to be fixed) than with established products. Not every Mallard customer will need support, but I am estimating that customer support will on average cost $6 per license for the first six months, and decrease to $3 per license after 18 months.

That is it for variable costs. Costs to operate servers, and payroll costs do go up with increased sales, however, on a per unit basis, the increase is relatively small. These costs will tend to stay constant until the number of customers gets to a level that more staff are required or more servers need to be set up, and then remain at the new level for some time. For the purposes of analyzing the potential profitability of Mallard, it is best to treat them as fixed expenses. As mentioned earlier, it is estimated that fixed costs will remain constant for the first six months that Mallard is sold.

Revenue:

The price range for a Mallard license is set at $30 - $50 for two reasons:

- This is what most anti-virus programs for personal use cost. There are ways to get protection from viruses for free, but the programs that cost money provide much better protection, and work quietly in the background without requiring user input. Millions of people are already accustomed to spending this amount of money for a useful computer program. A key difference with Mallard is that users only pay once for a license, instead of renewing the license on a yearly basis like they do with anti-virus programs.

- For a typical customer, it is pretty difficult to put a price on convenience. Customers are also going to be concerned about wasting their money on a program that doesn't perform as expected. Mallard will be sold with a 90 day no-questions asked money back guarantee, but customers will still feel like they are taking a risk by buying a Mallard license. For most people, $50 is their limit for "gambling" on a non-essential purchase that doesn't immediately make us feel better.

To provide an incentive for early adopters (the kind of people who are more likely to buy new, unproven products), the first 10,000 licenses will be sold for $30. Since upgrades are free for as long as Mallard is in business, early adopters have an incentive to buy sooner than later, since there is no reason to wait until a better version is available. The breakeven point of 500 licenses per month is based on $30 per license.

This proposal does not look at the financial aspects of selling information services to Mallard users, or selling software and services to businesses. Those sources of revenue should be self-supporting, at least in the long term, and the amount of revenue they produce will depend on Mallard being a successful product first.

Profit:

The following chart is a good test of your eyesight and shows projected net income for

24 months after Mallard is released.

		Licenses sold Monthly	Licenses sold Cumulative	Revenue per license	Variable costs	Monthly fixed Costs	Monthly net income	Cumulative net income
2010	September	30	30	$30.00	$10.00	$10,000.00	-$9,400.00	-$9,400.00
	October	60	90	$30.00	$10.00	$10,000.00	-$8,800.00	-$18,200.00
	November	120	210	$30.00	$10.00	$10,000.00	-$7,600.00	-$25,800.00
	December	240	450	$30.00	$10.00	$10,000.00	-$5,200.00	-$31,000.00
2011	January	360	810	$30.00	$10.00	$10,000.00	-$2,800.00	-$33,800.00
	February	540	1350	$30.00	$10.00	$10,000.00	$800.00	-$33,000.00
	March	810	2160	$30.00	$9.00	$10,000.00	$7,010.00	-$25,990.00
	April	1215	3375	$30.00	$9.00	$20,000.00	$5,515.00	-$20,475.00
	May	1823	5198	$30.00	$9.00	$20,000.00	$18,272.50	-$2,202.50
	June	2734	7931	$30.00	$9.00	$20,000.00	$37,408.75	$35,206.25
	July	3417	11348	$30.00	$9.00	$25,000.00	$46,760.94	$81,967.19
	August	4271	15620	$40.00	$9.00	$25,000.00	$107,416.02	$189,383.20
	September	5339	20959	$40.00	$8.00	$25,719.46	$145,139.92	$334,523.12
	October	6674	27633	$40.00	$8.00	$30,725.10	$182,849.11	$517,372.24
	November	8343	35976	$40.00	$8.00	$36,982.16	$229,985.61	$747,357.85
	December	10428	46405	$40.00	$8.00	$44,803.48	$288,906.23	$1,036,264.08
2012	January	13036	59440	$40.00	$8.00	$54,580.13	$362,557.01	$1,398,821.09
	February	16294	75735	$40.00	$8.00	$66,800.95	$454,620.48	$1,853,441.58
	March	17924	93658	$40.00	$7.00	$80,243.85	$511,243.59	$2,364,685.17
	April	19716	113375	$40.00	$7.00	$95,031.03	$555,605.15	$2,920,290.32
	May	21688	135063	$40.00	$7.00	$111,296.94	$604,402.86	$3,524,693.18
	June	23857	158919	$40.00	$7.00	$129,189.43	$658,080.35	$4,182,773.53
	July	26242	185162	$40.00	$7.00	$148,871.18	$717,125.58	$4,899,899.11
	August	28867	214028	$40.00	$7.00	$170,521.09	$782,075.34	$5,681,974.45

If you want to see the math behind this chart (or would like to see it enlarged so you can read it) I can send you the Excel spreadsheet it is based on. Even with a 10% increase in monthly sales after 18 months you can see what the effect is of continuous sales increases. The time frame in this chart is fairly optimistic, and realistically it could take three or fours years to reach the level of sales this chart shows for 24 months. Nonetheless, if Mallard is successful, it should eventually hit monthly sales of 25,000 to 30,000. That level of business is not sustainable forever, but it will be great while it lasts.

Sales of successful software tend to follow a squashed bell curve. Until the program becomes well known, sales are driven by word of mouth, and it takes a long time for the user base to build up to a point where the program becomes “the next big thing.” Percentage-wise, sales are increasing rapidly, but the number of units sold is still small. Then the general population becomes aware of the program, and large numbers of people want to try it out. The percentage by which sales increase begins to taper off, but the number of units sold in a month might equal the total number sold in the first year. Shortly after the program reaches the point of widespread brand recognition, sales start to gradually decline, because a growing percentage of potential customers already have the program. For Mallard, the pool of potential English speaking customers might be in the tens of millions, and if only 10% of those prospects end up buying a Mallard license, it still puts the program in the same ballpark as ACT!. As far as I can tell, there is no reason why Mallard could not be translated into any Indo-European language. Based on current trends in the software industry, translating Mallard into Spanish, French, Portugese, Russian and German will result in a market that is twice as large as sticking with just English.

The most profitable part of this cycle occurs when sales are relatively steady (as long

as steady is a big enough number). That is when support costs are the lowest, there are fewer bugs to fix, and less programming time is needed to keep it up to date. Selling software to businesses so they can run their own Mallard servers is one way to prolong that period of steady sales. Realistically, there is a limited number of people who will pay $50 for a PIM, if it cannot be used at work. On the other hand, if a computer program can save a company money by making its employees more productive, price is not much of an object. The business software market is overburdened with programs that deliver less than they promise, and usually the biggest cost is lost productivity, closely followed by support services, so the barrier to convert a business using Outlook is not insurmountable. Lotus Notes is perhaps the only product in this software category that is so expensive companies are hesitant to replace it with something else, because of the money invested in it, in terms of employee training and one time license fees.

Which is a good way to segue into another opportunity for an excellent return on your investment if Mallard gets to the point where it has good brand recognition. There are larger companies in the business software industry that could use Mallard as a way to gain a competitive advantage for the other programs and services they sell. Mallard will never command the price that YouTube did ($1.65 billion in Google shares), but a number of lesser known utility programs have been bought up by larger software houses for seven and eight figure amounts. I can always go back to being a sales rep.

The Nuts and Bolts of a Mallard Database

At its core, Mallard is a program that reads and writes text from and to an SQL (Structured Query Language) database. An SQL database has data stored in records (sometimes called rows) which are comprised of fields (sometimes called columns). Records are grouped together in tables. Data in one table can be linked or related to data in another table, so SQL can used to create relational databases. These days, virtually every complex database application uses SQL.

A Mallard database is also ACID compliant. ACID stands for Atomicity, Consistency, Isolation and Durability. The very short explanation of this is ACID is a guarantee that database transactions will be processed reliably. Therefore, if an attempt to add or change information in the database fails (power failure, program crashes, etc.), the rest of the database is not affected. Information is tested to make sure it fits the database structure before it is stored in the database and that information is not affected by other processes or programs while it is being read or written.

A Mallard database is encrypted with two passwords, one is used to initially create the database, and is embedded in the Mallard program, and the other password is selected by the Mallard user, so he or she is the only person who can open the database. When someone buys a Mallard license, they get a personalized (and encrypted) database that has their license number and a considerable amount of Public information already loaded into it. Mallard customers can pick and chose what types of Public information they want preloaded in their database,.

When a database is being used, and there is a connection to the Internet, Mallard checks with a Mallard server to see if any of the Public information in the database needs to be updated. The Mallard server also checks to see if there is a valid license for that database, and makes sure that only one database exists per license. The owner of a Mallard license could give out his or her password to others, but they would all have to work with the same database, and only one of them could work with the database at a time, which is an effective deterrent against piracy. The database file can be stored electronically anywhere; on the owner's computer, on a memory stick, or on a remote server, but without the password selected by the owner, no one can make sense of the information in it.

How information is processed

Allowing only one person to access a single database at a time has practical advantages. New information added to a Mallard database is based on (or related to) information that already exists in the database and all information is related to the owner of that database in terms of space and time. Every time data is added to the database, the local time and the physical location of the database owner is recorded. If there are changes to that relationship in time and space (the owner moves to a different city, for instance) Mallard can still determine the relationship that applied prior to those changes. It is actually possible to go back or forward in time and have Mallard properly manage those relationships. It simply isn't feasible to do that if information could be related to more than one owner.

Design principles behind Mallard

Without writing a book, there are three fundamental guidelines behind the design of a Mallard database:

1. Information requirements vary greatly between individuals, but for every individual, their own data tends to be repetitive and consistent.

2. Information which cannot be related or connected to an individual is not important to that individual.

3. No one can be in two places at the same time, but we can do more than one thing at the same time.

To meet the first guideline, a Mallard database has to be able to accept information with very few restrictions on how that information is formatted. The model of index cards was chosen for this reason, with the user giving a "title" to a piece of information that has meaning for the user, but not necessarily for anyone else, and then words and numbers can be added to the rest of the card in whatever fashion the user wants. Mallard searches the user's database for existing information that contain those words and stores references to that information in the content part of the index card. To add a new card, the user selects an existing card from the database of the type required (mailing address, grocery shopping trip, reading preferences, etc.) and makes changes to the card that was retrieved from the database. The user has to indicate whether those changes are to be used to make corrections to an existing piece of information or to add new information to the database.

The second guideline makes it possible for a Mallard database to store extremely detailed information in a relatively small file. It also makes it possible to perform validity tests on information that could not be done if the information had to be related to more than one person. Information in each scope in the database is self-contained so that the coherence of that information is not broken when changes are made to information in another scope. When sharing information between databases, information is never inserted directly into the other database, the owner of each database "accepts" information by relating it to existing information in her or her own database. Based on how the user has "accepted" shared information in the past, Mallard can do this with very little user input.

The third guideline allows Mallard to display time related information in a sensible way. Without this guideline, it would not be feasible to allow the user to set time constraints for Plans and if the user wanted to manage the details behind Procedures, it would not be feasible to check the information entered for consistency. Mallard needs to be able to recreate the state of the database for any given point in time, in order to check for consistency, and the time information necessary to do that is stored in the database.

Recycling Information

If you remember the three R's of recycling (reuse, reduce and recycle) a good analogy for Mallard is that it recycles our personal information. It reuses previously entered information whenever possible, it reduces the amount of information we have to enter, and one piece of information can be recycled into another piece, through child objects inheriting information from their parent objects. This is a concept that can be easily applied to how humans process information, but it is surprisingly difficult to do in a computer program. Computers have no built-in understanding of the meaning or context of information. Human beings have to "program" computers to "understand" the data that is being processed. Unfortunately for computer programmers, different human beings have different understandings of what a particular piece of information means, and despite the best efforts of some very smart people, we aren't any closer to artificial intelligence than we were fifty years ago.

I avoided this problem with Mallard by letting each user apply their own understanding of whatever information they want to keep in their own databases. If you have used Quicken or Quckbooks to manage your finances, you have already seen how effective just the "reuse" part of information recycling is. There is no artificial intelligence in Mallard, just new ways to manage information that is relevant to ourselves, but might not be relevant to anyone else. I have tried to make it as easy as possible to take information from other people (technically, other computers) and incorporate that information into our own databases, but once information is in a Mallard database, it really only makes sense to the owner of that database.

You might find it interesting to look at what happens when computer programmers try to write a program that "thinks" like humans. You could start your research with the Semantic Web and then check out two PIMs, Lotus Agenda and Chandler.

Technical overview of Mallard:
(no attempt is made to avoid jargon in this section)

If you are knowledgeable about SQL and relational database design, you are probably concerned about how much a Mallard database resembles an object-oriented database. I made a number of design choices to avoid the pitfalls of OO-relational databases, and now that I've written enough of the program to be able to create, read, update and delete objects in a demonstration database, I am confident that performance will not be a problem with Mallard, even with a relatively large database.

The database engine behind Mallard is SQLite3, which is outstanding at performing SELECT and UPDATE statements, as long as concurrent access is not required. Only one user can use a Mallard database at a time, so record locking and using threads for concurrent access is not an issue. A Mallard database has more than 35 stored triggers, and more than 20 defined indexes, which definitely slows down INSERT operations, but the number of records that need to be added at the same time is fairly low. Based on my results loading Public scope records into a new database, Mallard should still be able to insert 500 records per second on a Pentium 4, and users are not going to notice if it takes a full second to flush changes to the database, anyway.

After some failed attempts to design a system that used calculated or derived keys to place records in a hierarchy, I converted everything to nested sets. Time constraints for Plans are indexed with a two-dimensional R-tree, which is compiled into SQLite3. I spent a fair amount of time researching suffix and Patricia tries before I realized that redundant storage of text in the database was never going to be enough of a problem to justify adding a great deal more complexity to the program.

Without giving too much away, I have found that most complexity issues can be resolved by matching strings, because we process information with language. By making users relate their input to the displayed text for existing records, Mallard does not have to make those connections programmatically, which makes the program look more sophisticated than it really is. Mallard is designed to make it easier for users to provide context for the information they input, and use that context to infer context for other data. I was not able to use datasets and data-aware controls, because the information displayed for the user cannot be directly mapped to fields and records in the database itself, but that allows for greater flexibility in accepting user input. I doubt if this approach can be generally applied to database applications, but for the purposes of a PIM, it seems to work quite well.

Disclaimers and important notes regarding the preceding document:

- This is not a prospectus, it is a speculative document that provides information about an incomplete computer program still under development. The purpose of this document is to gauge the interest of potential investors in funding further development so this program (called “Mallard” in this document) can be brought to market.
- The advice of legal and accounting professionals has not been utilized in the preparation of this document. Any statements of fact, forecasts or comparisons in this document could be inaccurate or false.
- This document is confidential and for private use. Registered trademarks and copyrighted content may be included without attribution and without permission from the owners of those trademarks and the holders of those copyrights.
- Rick Glasel holds all copyrights to this document, the computer program that this document refers to, and any other documents or programming code associated with this program. Mallard is not a registered trademark for this program. .