Creating an App Spec

How to prepare for your first mobile app!

by Mark Larson
Marblesoft, LLC

MARBLESOFT

Welcome!

Thank you for choosing Marblesoft as your app developer. Creating your first mobile application can seem a daunting task. It's really not as difficult as you might think, as long as you break into manageable tasks. This document will help you do that, and we're here to help.

We'll describe the process for preparing to develop a mobile app to run on iOS or Android devices. This process would be very much the same for a Web app, or even a desktop app for Mac or Windows. In fact, you can use this worksheet to prepare for any type of application. Just ignore the parts that wouldn't apply in your case.

This document is broken into two sections. The first, Ideas and Objectives, is where you'll tell us all about yourself and the basic ideas for the application. In the second, Designing the Specification, we'll get into the details of exactly what the app is supposed to do.

Feel free to contact us with any questions as you go through this process. You may have already established a relationship with one of our staff. If so, they'll be happy to assist you in any way. If you have don't already have a Marblesoft representative, please let us know, and we'll hook you up.

Marblesoft Development Team
12301 Central Ave NE Suite 205
Blaine, MN 55434

info@marblesoft.com
763-502-0440

Contents

Creating an App Spec

Part One

Ideas and Objectives

MARBLESOFT

In order to give you a flat rate bid, we need to know exactly what the application is designed to do. In this first part of the process, you'll just be giving us the basic information on who you are, what kind of application we'll be making, and what the application will do.

1.1. Company description

Company name ______________________________

Address, City St Zip ______________________________

Main contact person ____________________ Phone ____________________

Contact person's e-Mail ______________________________

1.2. Application description

Project name (working title) ______________________________

What kind of app will this be? Mark all that apply, even if they won't be done until a future version. That will help us plan and save you money long term.

□ iPhone/iPad □ Android □ Web-based □ Macintosh □ Windows

What's your budget for the first version of the app? ○ $0 - 5,000 ○ $5,000 - 20,000
○ $20,000 - 50,000 ○ $50,000 - $100,000 ○ Over $100,000

What's the deadline for the first version? __________ If there's a specific event like a tradeshow or the busy season for your business, list the details. ______________________________

When would you like to start? ______________________________

1.3. Design phase

Before we can begin programming your application, we need to know exactly what it is supposed to do. The design phase involves three main areas, look and feel, user interface and functionality. The look and feel is the basic visual design like logos, splash screens, whether it appeals to adults or children, etc. The user interface is the actual screens, buttons, menus, lists and messages that define how the user works the program. The functional design is determining exactly what the program will do and keep track of.

Will you design the app's look and feel, or do you need Marblesoft to design it for you? Please describe in detail.

____

____

____

Will you design the user interface, or do you need Marblesoft to design it for you? Please describe in detail.

____

____

____

Will you design the app's functionality, or do you need Marblesoft to design it for you? Please describe in detail.

____

____

____

1.4. App details

Write a general a description of the entire program and what it does. Feel free to revise this description as you think of new things that should be covered. Use more pages if you need them. Every hour you spend on this description will save you hundreds of dollars of programmer and illustrator labor. The more precise you are, the less we have to guess.

(Write description here)

1.5. Resources

Many apps require resources like images and sounds which need to be acquired. We may be able to use pre-existing resources you already have, or we may need to license them from a third party or even create them from scratch.

You may also need to license data from a third party, such as a dictionary, a map library or perhaps an RSS feed. We'll need to know where to get such information, and the more you can source for us, the less we'll have to charge you.

Make a detailed list of all the images, sounds, movies, Web databases, RSS feeds and other resources you will need for the app. Next to each item, mark "have", "license" or "new".

(list needed resources here)

1.6. Funding

OK, we have to think about this. There are several ways you can fund a mobile app. When you figure out which one (or more) will work for you, you're ready to proceed. Listed below are some of the common methods, along with our thoughts on them.

1. The App Store. There are two basic ways of making money on Apple's iTunes Store or on the Google Play (Android) store. The first is the actual price of the app, and second is in-app purchases for additional content. In case you didn't know, Apple or Google takes 30% off the top of every app sold and every in-app purchase, too. There is no charge for hosting free apps. In that case, you'll have to cover development costs another way, which is what the rest of this list covers.

2. Internal use. Some apps, whether they're sold on the App store or not, are designed to save your company labor or to create other efficiencies in your own operation. You will be able to justify the expenditure for what the app will do for your business, and then selling it on the app store is just a bonus.

3. Product promotion. Your app could be designed to help sales of a physical product. The app is paid for by the sales of that product that it promotes.

4. In-app advertising. This pays off if your app is seen by large numbers of people. In-app advertising may not be appropriate in apps for certain audiences, like education.

5. Sponsorship. Your app could be funded by a third party because it promotes their interests. Apple and Google have certain restrictions regarding this. This could be a great strategy for an app like a fund-raiser. You can donate to the charity by using this app provided by the good people at such-and-such.

Please specify how you plan to fund the app using these methods or any other ideas you have.

(list funding sources here)

That's the end of Part One. When you've decided on exactly what you're app is going to do, go on to Part Two.

Creating an App Spec

Part Two

Designing the Specification

MARBLESOFT

This document assumes you have already completed Part One, Ideas and Objectives. Now we can start designing the actual application. Have fun with this!

2.1. Screen Layouts

The first thing is to design the screen layouts. You don't have to draw every animation that will be needed, but you need to draw every screen that will appear in the program. Make it as plain or as fancy as you like, but make it detailed - all the text (or placeholder text) that will be needed, and placeholders for each graphic element. Scribbles and stick figures are enough. You'll work with an illustrator on the detail later. For now the important part is that you think about and show everything you'll need. As usual, failure to flush out everything at this stage will result in additional labor costs later.

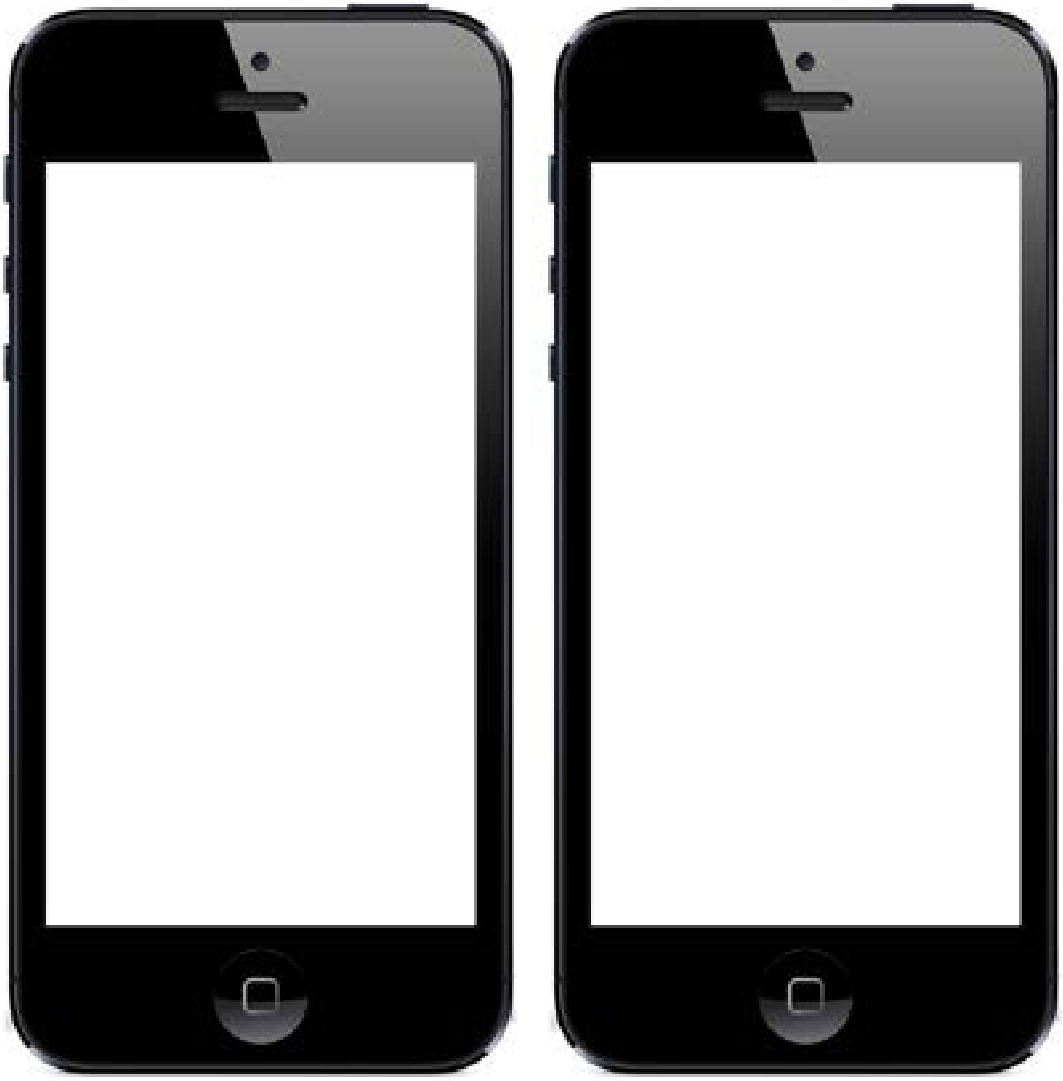
Copy this page as many times as you like and use it to draw what the program will look like on a phone. If it will look different in landscape and portrait mode, use a separate drawing for each. Just turn the page sideways.

Screen:

Description:

Screen:

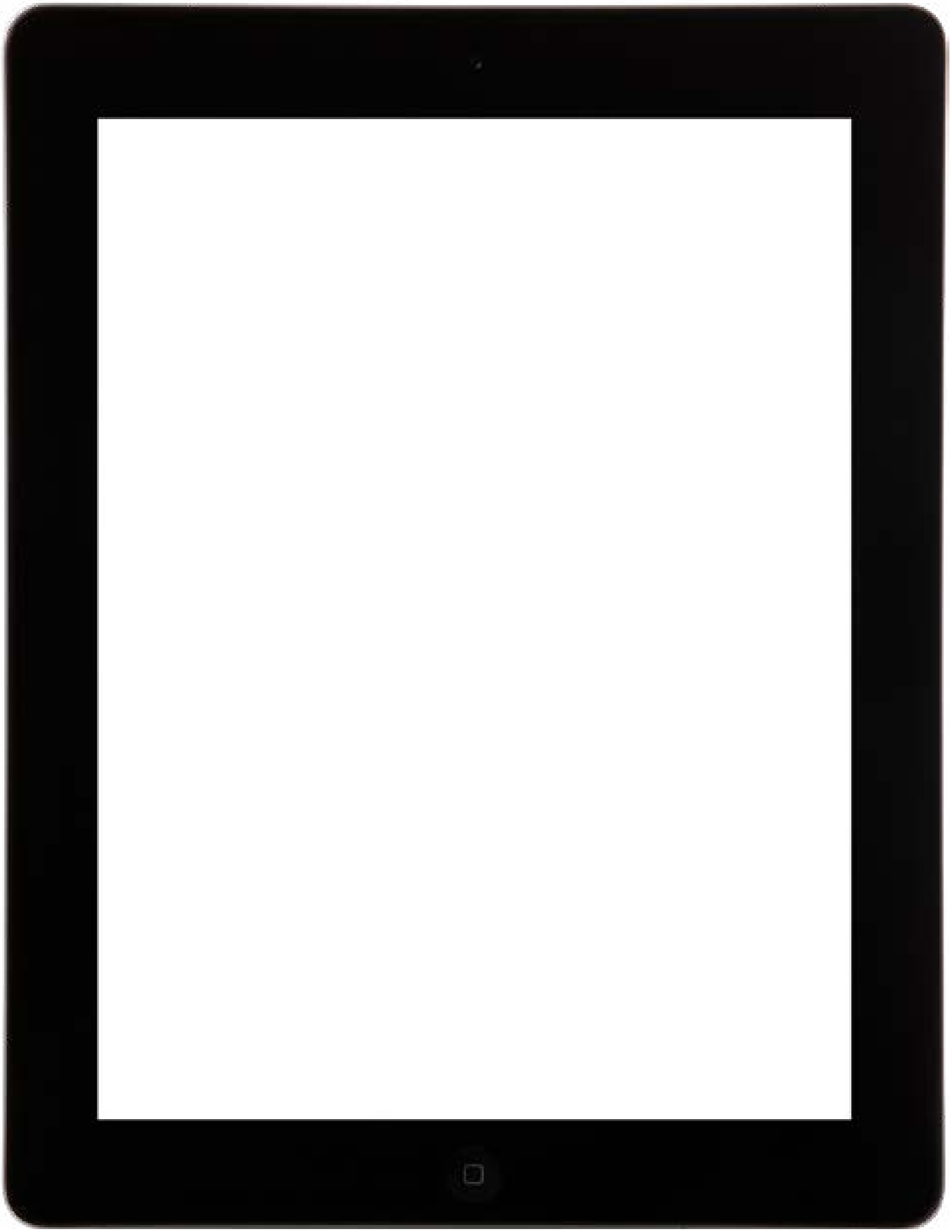
Description:



Copy the following page as many times as you like and use it to draw what the program will look like on a tablet. Specify whether this screen will run in landscape mode, portrait mode, or both. If it will look different in landscape and portrait mode, use a separate page for each. Just turn the page sideways.

Screen:

Description:



Screen:

Description:

2.2. Roles (optional)

List all the different types of users who might use this application. For instance, you might have an administrator who can set up accounts for other users. You might have a regular user, or perhaps a customer user. Most apps usually only have one role, but if you have more, you need to document what each role is.

After each user, specify what functionality that user will have, and exactly which screens each type of user will be able to access. It's possible you'll need more than three types of users, but if you do, you might want to rethink your design.

User:

Role:

Screens:

User:

Role:

Screens:

User:

Role:

Screens:

User:

Role:

Screens:

2.3. Data Structures

This is the hardest part of the spec. We can help you with the details, but the more you can think of in advance, the less it will cost you in the end. You may only have one or two pieces of data to record. Likely you will have dozens or more. A complex program might require dozens of tables with perhaps dozens of fields.

The first decision is what data will be saved (called persistence), if any. An app like a news reader would not save any data. A game might save your scores. A utility might save everything you do. You need to figure out what data will persist, and what will be lost when the user quits your app.

Second, you must decide where the data will be stored. Will it be stored only on the user's device? Will it be transferred to other devices or some other permanent storage entity (like the "cloud")? As usual, no data storage is the cheapest, and the more broadly you store and share your data, the more expensive it becomes.

Here's a little questionnaire to help you figure out your storage requirements.

1. Will the device store any information, as opposed to losing everything when the user quits? ______________

2. Will anything need to be stored on a remote device or server? ______________________________
Specify ____
____

3. Will the user be able to select and send certain data to another user? ______________________
Specify ____
____

Your data will be stored in files in pieces called tables. Each table contains all the records of a certain type. Each record has a number of fields describing the entity. You need to document every field that will need to be stored. Fields are most typically either text or numbers. Text is anything you can type on the keyboard. Numbers are only numerals and decimal points describing the number. For instance:

Text	**Numbers**
Apple	1
(763)555-1212	0
1+ 5 / 4	-7635551212
$1.99	1.99

Most records need to be identified by an ID number. When records might be created by diffeent users, instead of a number we use a unique identifier that can only be created on one device at the exact moment in time. That way no two records can possibly have the same ID. This unique identifier is called a UDID in the table. If you want an ID that's more human-readable, like an invoice number, make it a number or text.

There are other types of fields you may need, like a picture or sound file, an array, which is a list of objects, and raw data, like a stream of measurements from an instrument. If you understand these types of fields, include them in your spec. Otherwise, we'll help you with that when we get to that point.

Organize your data in tables describing each entity to be saved. Within each table, define the fields that will store a detailed description of the entity.

On the following page is a very simplified example of a data structure. Yours will likely be much different.

Table: Employees

Fields (name and type):

Employee ID	UDID
First name	Text
Last name	Text
Address	Text
City	Text
State	Text
Zip	Text
Hourly rate	Number
Photo	Image
Department	Number

Table: Departments

Fields (name and type):

Department ID	UDID
Department name	Text

Table: Products

Fields (name and type):

Product ID	UDID
UPC	Number
Product name	Text
Price	Number
Photo	Image

Table: Sales

Fields (name and type):

Invoice ID	UDID
Invoice number	Number
Customer ID	UDID (You'd have to define a customer table, too, though we haven't here)
Invoice number	Text (may include hyphens or letters, etc.)
Date	Date
Employee	Number
Products sold	Array
Sales tax	Number
Total paid	Number

Table: Product sold

Fields (name and type):

Product sold ID	UDID
Invoice ID	UDID
Product ID	UDID
Quantity	Number
Price	Number (you record the price here because it might be discounted for this order)

Use the form on the next page to start building your data structures. Use as many pages as you need.

Table:

Fields (name and type):

Table:

Fields (name and type):

Table:

Fields (name and type):

Table:

Fields (name and type):

2.4. Phases

Now for the fun part. List every feature on your wish list. Don't leave anything out. If it would make your app better, put it on the list. Once you've run out of ideas, make a second pass at the list. Lightly mark in pencil all the must-have features, the things you can't possibly release an app without, with a 1.

Look at your list again. With everything you've marked with a 1 built into the app, do you have a saleable product (or if it's for in-house use a useful product)? Some people call this the Minimum Viable Product, a term I'm not particularly fond of, but it has a great acronym - MVP. The MVP is what we want to do in the first version. If you try to do everything in the first version, it will take longer, and you won't make any money with it for a long time. With an MVP, you can start getting a return on your investment as soon as possible, and that payback is what will fund the rest of the project as an upgrade later.

Finally, go through the list again and finalize the features for version 1. Mark the version 1 features clearly, and mark each other item with a 2 or 3 to indicate in which version the feature will be done. When you're done, you will have a complete specification that shows us what we need to do to properly estimate the cost of the project, and will allow us to work quickly and efficiently.

Version 1 Features

(list features here)

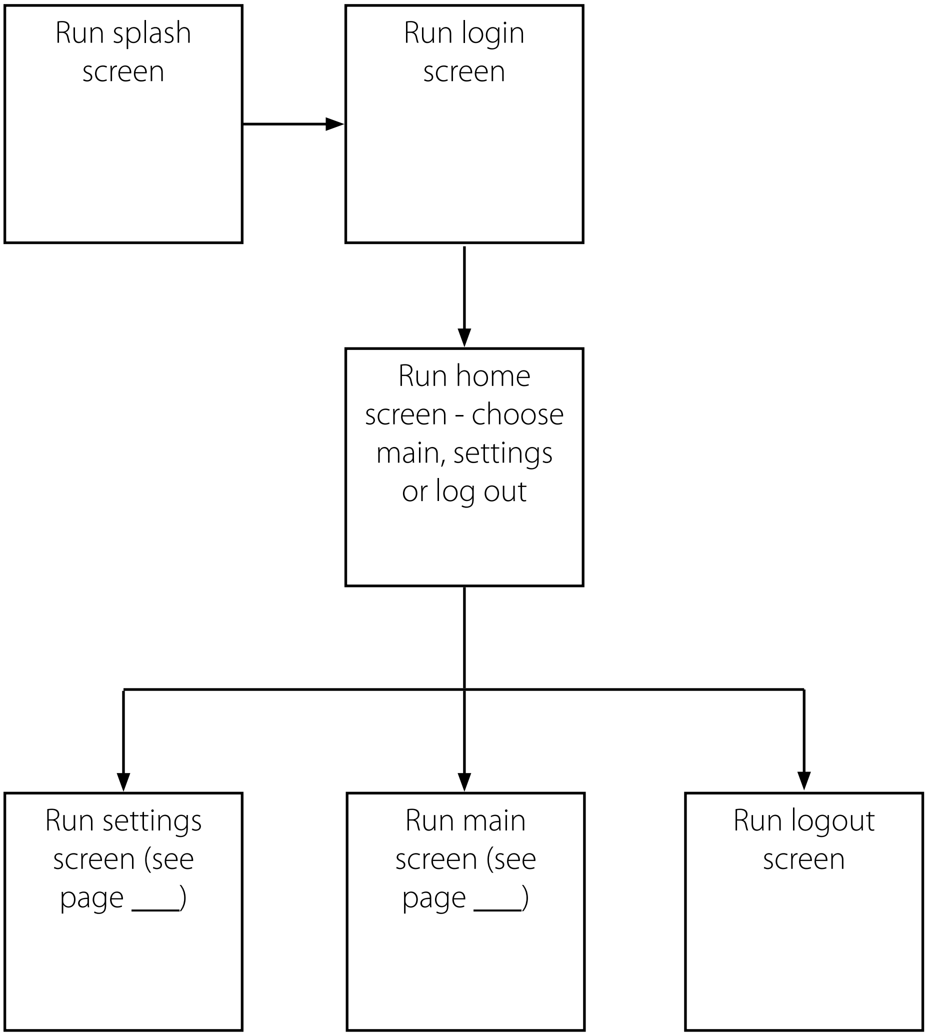
Features for Later Version

(list deferred features here)

2.5. Flowchart

Draw a flowchart of the basic functionality. For a complex program, break the functionality into groups, with each group on its own page. Make as many pages as you need to define the complete functionality. Again, every minute you spend here will be several minutes a programmer doesn't have to spend trying to guess what you intended.

Below is a sample flowchart for the login/home screens of a simple application. The flowchart style doesn't matter. Just make it easy to follow. In this case we're only showing the connection of the screens and a simple branch, where the application has to make a decision based on user's choice from the home screen.


Run splash screen
Run login screen
Run home screen - choose main, settings or log out
Run settings screen (see page ____)
Run main screen (see page ____)
Run logout screen

2.6. The process

Write up this app specification to the best of your ability. Remember, any part of the spec that you don't do, you'll need to pay us to do. The more accurately you can describe the app, the more accurately we can calculate the cost of it, and the less it will cost you in the long run.

Once you've designed the spec, we'll give you a cost estimate. If you give us a complete spec, we can usually give you a guaranteed bid. If you give us a vague spec, we won't be able to give you a flat rate, and more of the app will have to be billed hourly.

As we go, we'll show you our progress at regular intervals. You'll be able to approve the design, and we'll be able to make changes if we discover that the original design had flaws. Watching the app come to life is a very rewarding process, and I'm sure you will enjoy it.

We look forward to working as part of your team.

Mark Larson
Marblesoft
12301 Central Ave NE
Suite 205
Blaine, MN 55434

763-502-0440
mark@marblesoft.com

(put additional comments here)