

Process Description

Software Toolset

As our product will be an Android application, the primary programming language that will be used is Java, which is the standard for Android development. The primary data sources are user driven, meaning that users create their own lists, but we might also rely on Google maps for location data. Given time, Yelp might be used to feed additional data to our application, such as restaurant, movie, or store information. GitHub will be used as software version control, and the integrated issue tracker will serve to track bugs. The wiki on GitHub will act as a place to document the project. Google Documents will be used on occasion when the need arises for multiple parties to edit the same documents, although this will be used primarily for assignments or reports and not code. It early to tell what “off the shelf” software components we will need, but at first glance, we will certainly use the Android SDK, and possibly later we will employ the use of Google Maps API or the Yelp API.

The reason we are choosing this toolset is to minimize what we have to implement from scratch. Android is the platform of choice for mobile applications, and the social aspect of our project lends itself well to a mobile platform.

Group Dynamics

The project manager was originally Matthew, who dropped the course after week two. At the upcoming team meeting, a new project manager will be elected and the course staff will be notified. The team structure is fairly resilient, however, and the team should be able to transition easily over the next week. Matthew has agreed to complete his share of the SRS Requirements and ensure the team meets this milestone. All relevant documents will be passed on to the remainder of the team, and Matthew will be reachable via email should any issues arise. (Former PM’s note: this is a great team, and I have no doubt they will succeed! –Matthew)

Over the remaining eight weeks of the course, the team will flush out more exact roles. Initial discussion revealed each team member’s initial interests: Troy said he was interested in database programming; Colleen mentioned she did not want to use Swing, but liked HTML and model view style and wanted to work in the middle or back end; Lawrence mentioned having had published an Android application and is willing to do PHP to server to database connection programming; Will also has Android development experience and has volunteered to do the graphical front end; Davis expressed interest in the low level and back end part of the project. Based on these interests, the team might decide to divide the programming load into specific areas of responsibilities and assign corresponding roles, such as Lead or Design Developer (to coordinate the team’s development efforts), Database Developer (one or two team members), and User Interface Developer (Will). These roles have been decided based on each person’s individual interest—people tend to work better if they are interested in what they are doing.

The roles might differ for different parts of the project, for example, if time arises and we are able to delve into the Google API or the Yelp API, we might need to rethink who will be responsible on this new frontier. Initially, however, the team will prefer to have clearly defined programming roles so that components from each developer can be easily integrated.

The team has not yet explicitly established how disagreements will be resolved, however the system in place now is implicitly a combination of communication and voting. To be specific, our official team name is currently undecided. An email was sent to the group email list, but there was no clear consensus. The team agreed to discuss this issue in our group meeting this Tuesday and resolve the issue by voting. In short, our team will resolve disagreements by first talking about it as a group, and if we do not have time in a meeting to resolve all differences, we will hash it out over email. Email is our preferred means of communication second only to meeting as a group, so email will likely be the vehicle for working out issues. We value our differences and are motivated to synergize as a team.

Schedule / Timeline

Role / Sub-Group	Feature to Implement	Expected Time Cost	Completion Date (estimate)
Design Developer	• System architecture	10 days	30 April
	• Design of public interfaces	5 days	24 April
	• Integration diagrams	5 days	1 May
	• Test cases	5 days	8 May
Database Developers	• Entity-Relationship models	5 days	17 April
	• Database creation	7 days	26 April
	• Common SQL prepared statements	8 days	3 May
	• Items can have multiple tags	5 days	10 May
	• Query db for category, name, or tag in user list	3 days	15 May
UI Developer	• Map all items of category/tag	5 days	17 April
	• Make notes on list items	5 days	24 April
	• User chooses how to categorize items	5 days	1 May
	• Map items with address with colored pins	15 days	15 May

The original idea for ForgetMeNot (code name Android Lunch) was Colleen's, so she may choose to be the Design Developer. The Design Developer will be responsible for the overall system architecture and specify how the various parts fit together.

Risk Summary

Learning Curve: Learning a vastly new API such as the Android SDK a challenge.

Feature Creep: We may find ourselves wanting to add extra features so the application is functional, yet we might not have the time.

Integration: Integrating our product smoothly with other applications such as Yelp or Google Maps will be a challenge.

To mitigate these risks, we will devise a system of checks and balances to determine whether features are essential or auxiliary. We will strive to add only essential features until a minimal, but functional, system is in place. Integrating with other services might prove easy (such as for Google Maps) or difficult (such as with Yelp), but because we do not know for sure, we will spend only a limited time researching, as we do not want to waste too much development time doing research that might not pay off.

We will create a user interface “scaffolding” that displays each of the possible options and features that we plan to implement. This scaffolding is simply a bare-boned user interface including buttons and windows, none of which will be functional in the beginning. As we add features, the buttons will be enabled to trigger the added functionality. If we cannot get certain components to work on time, we will simply display a message box informing the user that the feature is planned for a future release.

About two weeks prior to the beta release milestone, we are planning to contact our customers and clients and gather user evaluations and input through observing how the user navigates the application. We will provide little or no instruction to the users to test whether our interface is intuitive. Additionally, we will solicit for feedback on how to make a better product and what areas need attention.