INFOmap

Indoor Navigation For the Optically-challenged

Project Part II: Specification

February 22, 2010

University of Nevda, Reno
Department of Computer Science
Instructor & Advisor: Eelke Folmer

Team 01:
Kevin Grant
Angela Proffitt
Alex Tam

Table of Contents

1 Abstract

INFOmap is a handheld navigation software primarily focused on helping blind patrons navigate the interior of buildings. It utilizes voice command and crowdsourcing to build a database of objects within a building. A map of the building can be downloaded off the internet and the user can then enter key items such as vending machines, restrooms, and chairs through voice and photo. Since the information is collected from a variety of users, the data will be compared to other entries to confirm the object and its location to prevent false information. As more objects are entered, the more accurate and fully navigational the area will become. While INFOmap is focused on helping the blind, it can also be used for any user unfamiliar with a building.

2 Introduction

INFOmap is a handheld navigator. The project is an extension of a cell phone navigation application Kevin Grant and Alex Tam have been developing for Dr. Eelke Folmer and IBM. The application is designed to help blind patrons and users unfamiliar with specific areas navigate and find objects and destinations.

Users can map anything, such as a park or building, by running the application while walking. A map of the area is uploaded onto the phone via internet connection through the phone or by connecting to the computer. Once the patron has a map of the area, the user will then be able to enter items into the INFOmap database. The user will choose a room to enter items. If the user is in non-blind mode, the user can point at an area of the room and enter the object by voice command or by selecting the object using the 'selection option'. If the user is in blind mode, the user can simply add an object into the room by voice command. The specific location within the room will not be handled. The user can also delete objects in the room by voice command or by selecting the object by pointing at it and deleting the object. At a later stage of the project, the user can also take pictures of the objects to create a compiled image of the whole room.

The main characteristic of the project is that INFOmap uses "crowdsourcing" to collect data. Crowdsourcing is the practice of outsourcing work to a group of people, usually in the general public, that is interested in the work. Anybody running INFOmap can enter objects into the database. The entries will be compared to confirm location and name accuracy for each object. The idea is to get as many entries as possible, thus making the database more accurate and filled with objects. Once INFOmap is filled with objects such as restrooms, vending machines, and chairs, a blind user or someone unfamiliar with the area can ask the phone where a certain object is. INFOmap will then navigate the user to the desired objects by giving visual and audible directions. INFOmap will be using the Android platform, which is based on Java.

Blind patrons often have a hard time finding rooms in an unfamiliar building. Using INFOmap, he/she can ask the phone where room A is, and INFOmap will give audible directions to room A making traveling for the blind much easier and less stressful. INFOmap is useful for anybody; a user no longer has to roam around the building and ask other people where the nearest restroom is. Another useful application would be to navigate through a theme park. The user can ask where a certain ride is, and INFOmap will direct him/her to the ride.

One of the biggest challenges will be to control crowdsourcing. Having unreliable sources may mean unreliable data. The problem is prominent in crowdsourcing websites such as Wikipedia. Since anyone can contribute to data entry, there can be false or mistaken information entered into the database. We will have to develop a method to compare entered data to confirm accuracy. We will also have to figure out in which manner to hold data and map out the buildings. Will the coordinates be taken where the data is entered or will user enter coordinates? If the coordinates are taken where the user is standing, will the person have to be on the object? With multiple photo entries, which photo will be chosen to use for mapping? What if the photo is unrelated, how will we detect its accuracy? How can we motivate people to enter data into the database? With no entries, crowdsourcing will not work. If the user is in a loud room, how will voice detection work? Will we need to implement manual entry? With a project this big, we are bound to run into many problems.

Beyond CS 425, INFOmap can be developed to become the largest database of directions. All the buildings will be connected in a full scale world-wide navigation similar to Google Maps. However, INFOmap can also navigate inside buildings. It can also become a social network application to find friends, lovers, or reconnect with lost family members by adding even yourselves into the database, not just objects.

Professionally, this project will expose all the team members with programming on a mobile device. Many companies are interested in crowdsourcing and mobile programming, and thus will be excellent to have on a résumé. Since this project has many social uses and has the potential to grow into a full scale application, it has the potential to become a grad thesis or even start a small company.

3 Software Specifications

3.1 Functional

LEVEL 1 - REQUIRED

FR1.0 The user shall be able to view a specific room for gathering information.

FR1.1 The user shall be able to record items and their locations to the map.

FR1.2 The user shall be able to create textual notes.

FR1.3 The user shall be able to create audio notes.

FR1.4 The user shall be able to extract textual notes.

FR1.5 The user shall be able to extract audio notes.

FR1.6 The user shall be able to view a help page from within the program.

LEVEL 2 - PRIMARY GOALS

FR2.0 The program will draw a birds eye view of the room on the screen.

FR2.1 The program will draw room items in specified locations based off of data in the KML file.

FR2.2 The program will have easily identifiable icons for text and audio notations.

FR2.3 The program will not send any personally identifiable information to the database.

FR2.4 The user will be able to select a point in the room and add an item note.

FR2.5 The user will be able to select an item in the room to view or change its note.

LEVEL 3 - SECONDARY GOALS

FR3.0 The program may have the ability to move and zoom in on the map.

FR3.1 The program may be able to change rooms easily.

FR3.2 The program may be able to receive picture input of an item for submission to a database.

FR3.3 The program may audibly announce specified items in a room.

3.2 Non-functional

LEVEL 1 - REQUIRED

NFR1.0 The program shall be written in Android (Java).

NFR1.1 The program shall handle simple voice commands .

NFR1.2 The program shall use a touchscreen interface.

NFR1.3 The program shall provide voice feedback.

NFR1.4 The program size shall remain under 2 Megabytes.

LEVEL 2 - PRIMARY GOALS

NFR2.0 A database of models will be hosted on the internet.

NFR2.1 The program's source will compile in Eclipse without being edited.

NFR2.2 The program will use an accelerometer and a gyrometer to detect user and object locations.

NFR2.3 The application will update note nodes every 1 to 2 minutes.

NFR2.4 The program will have an option to use a completely audible interface.

NFR2.5 The user will be able to speak into the device to control the program.

NFR2.6 The program shall have separate interfaces for sighted and blind users.

NFR2.7 The program shall store the notes directly within the KML model.

NFR2.8 The program will properly handle incoming phone calls.

LEVEL 3 - SECONDARY GOALS

NFR3.0 The program may run on other Android devices.

NFR3.1 The program may automatically install the text-to-speech library upon installation of the program.

NFR3.2 The database may be publically accessible.

NFR3.3 The programs source may be able to compile and run independent of Eclipse.

4 Use Case Modeling

4.1 Use Case Diagram

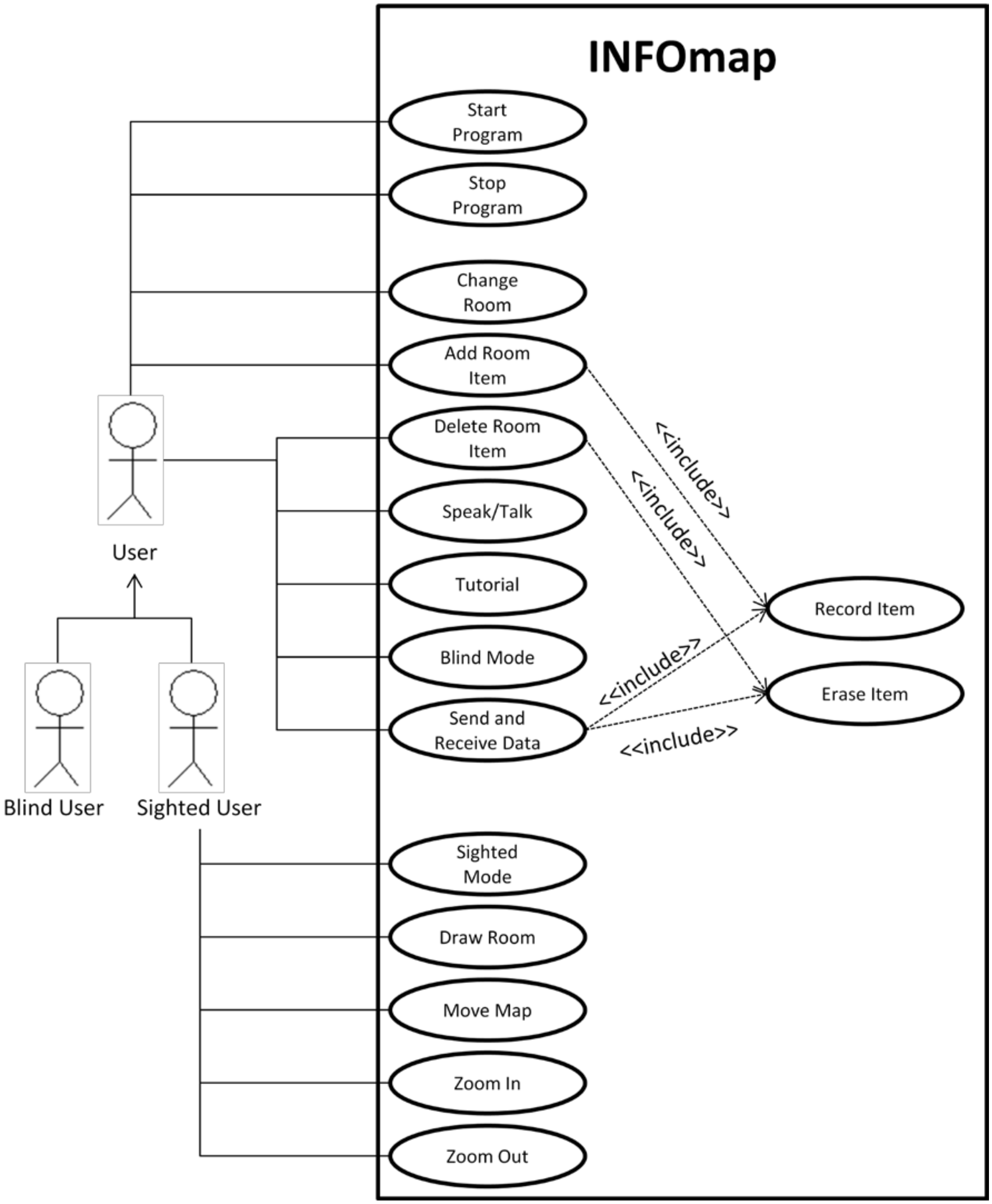


Figure 1: Use case diagram.

4.2 Detailed Use Cases

User

Main user of program

Sighted User

Inherits from User. User with the ability to see.

Blind User

Inherits from User. User who cannot see.

UCU1: Start Program

Begin running INFOmap executable on the Android operating system. Usually will run on a smart phone.

UCU2: Stop Program

End running of INFOmap through the use of "Quit" option. Android will allow quitting of the program via "Home" and "Back" button.

UCU3: Draw Room

A map of the room will be displayed on the device when ready. The map will be scaled to fit on the device's screen.

UCU4: Move Map On Screen

The map of the room can be moved around on the screen via touchscreen. The map will be limited on the edges from going off the screen.

UCU5: Change Room

The map of the room can be changed out with maps of other rooms. The option to change it can be activated via the program's menu.

UCU6: Zoom In

The user has the ability to zoom in on a point on the map. The option can be activated by a button on the screen.

UCU7: Zoom Out

The user has the ability to zoom out of a point on the map. The option can be activated by a button on the screen.

UCU8: Add Room Item

Room items such as furniture can be added to the server's database. The option can be activated by a menu option. The user can add an audible note and the option of a textual note.

UCU9: Delete Room item

Room items can be deleted from the server's database. The option can be activated by a menu option.

UCU10: Speak/Talk

Audible announcement of room objects will be a menu option in Sighted mode. The audible announcement will be turned on automatically in Blind mode.

UCU11: Tutorial

Tutorial is a special mode to help guide the user on how to use the program. The option for the tutorial will be available at the main screen.

UCU12: Sighted Mode

Sighted mode allows both auditory and visual feedback. The option to activate the mode will be available on every screen.

UCU13: Blind Mode

Blind mode allows users to just have auditory feedback without visuals. The option to activate the mode will be available on every screen.

UCU14: Send and Receive Data

Automatic synchronizing of data with server database. Will only transmit relevant data and no personal information.

UCU15: Record Item

Makes a new entry in database of room items. Adds the user-submitted data to the new entry. Included in “Add Room Item” and “Send/Receive Data”.

UCU16: Erase Item

Delete an entry in the database. Next time user accesses database, the item won't be there. Included in “Delete Room Item” and “Send/Receive Data”.

4.3 Detailed Use Cases

Use Case: Draw Room
ID: UCU3
Actors: User
Preconditions: 1. Previous screen was main menu or loading screen. 2. UCU1 completed.
Flow of Events: 1. The use case starts when the program needs to display a map either by asking the user for a room or loading the last used room from previous session. 2. Get the coordinates of the room from the large map of building. 3. Draw the room using the coordinates and scale the drawing to fit on the device screen.
Post Conditions: 1. Map of room is drawn.

Use Case: Add Room Item
ID: UCU8
Actors: User
Preconditions: 1. Map is loaded (UCU3 completed). 2. Item doesn’t exist yet.
Flow of Events: 1. The use case starts when the user activates the “Add Item” menu option. 2. The user touches a spot on the room’s map corresponding to the spot the physical item is located. 3. A dialog box will appear to allow the user to select an audio or textual tag or label for the item at the specified location. 4. The program will submit the data to the server’s database when the user hits the “Submit” button. 5. The room’s map will update with a new item in the specified location.
Post Conditions: 1. Item is labeled. 2. Item added to server’s database. 3. Item is visible on device’s screen.

Use Case: Delete Room Item
ID: UCU9
Actors: User
Preconditions: 1. Map is loaded (UCU3 completed). 2. Item exists in database.

3. Item may have been physically removed from room or item is in wrong location.

Flow of Events:

1. The use case starts when the user activates the “Delete Item” menu option.
2. The user touches an item on the map that should be removed.
3. A dialog box will appear to allow the user to confirm deletion.
4. The program will submit the data to the server’s database on confirmation.
5. The room’s map will update with the item removed.

Post Conditions:

1. The item is removed from the database.
2. The item is removed from the map on the device’s screen.

5 Requirement Traceability Matrix

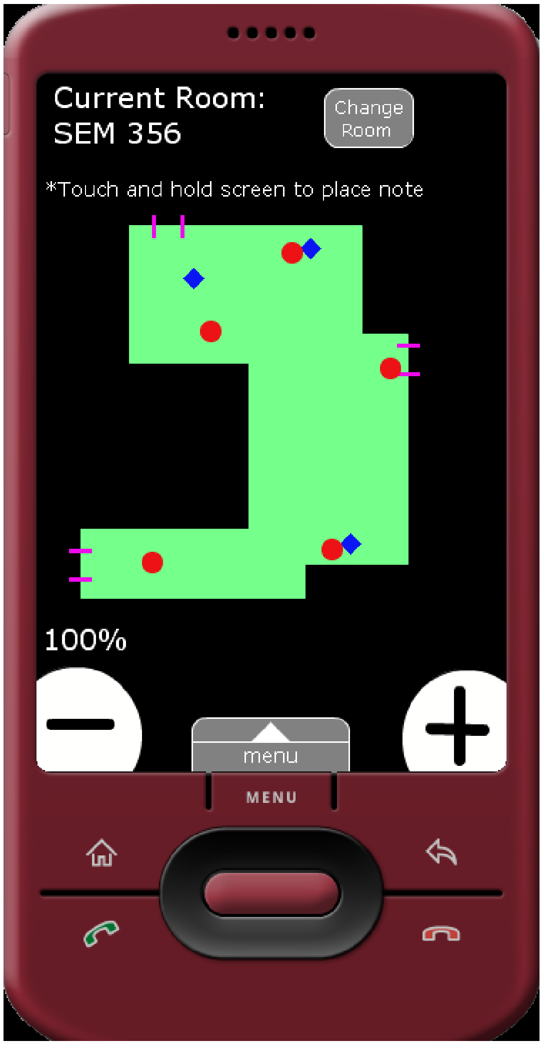
	UCU1	UCU2	UCU3	UCU4	UCU5	UCU6	UCU7	UCU8	UCU9	UCU10	UCU11	UCU12	UCU13	UCU14		UCS1	UCS2	UCS3
FR1.0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩		|			
FR1.1	🟩	🟩	🟩					🟩	🟩		🟩	🟩	🟩	🟩	|	🟩	🟩	
FR1.2	🟩	🟩	🟩					🟩	🟩		🟩	🟩		🟩	|	🟩	🟩	
FR1.3	🟩	🟩	🟩					🟩	🟩	🟩	🟩	🟩	🟩	🟩	|	🟩	🟩	🟩
FR1.4	🟩	🟩	🟩					🟩	🟩		🟩	🟩		🟩	|	🟩	🟩	🟩
FR1.5	🟩	🟩	🟩					🟩	🟩	🟩	🟩	🟩	🟩	🟩	|	🟩	🟩	
FR1.6	🟩	🟩	🟩					🟩	🟩	🟩	🟩	🟩	🟩		|			
NFR1.0	🟩	🟩	🟩							🟩	🟩	🟩	🟩	🟩	|			
NFR1.1	🟩	🟩								🟩	🟩	🟩	🟩		|			
NFR1.2	🟩	🟩	🟩		🟩						🟩	🟩			|			
NFR1.3	🟩	🟩			🟩					🟩	🟩	🟩	🟩		|			
NFR1.4	🟩	🟩	🟩								🟩	🟩	🟩		|			
FR2.0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩			|			
FR2.1	🟩	🟩	🟩					🟩	🟩		🟩	🟩			|	🟩	🟩	
FR2.2	🟩	🟩						🟩	🟩	🟩	🟩	🟩			|			
FR2.3	🟩	🟩						🟩	🟩		🟩	🟩		🟩	|	🟩	🟩	🟩
FR2.4	🟩	🟩	🟩					🟩	🟩	🟩	🟩	🟩		🟩	|	🟩	🟩	
FR2.5	🟩	🟩	🟩		🟩			🟩	🟩	🟩	🟩	🟩		🟩	|	🟩	🟩	
NFR2.0	🟩	🟩									🟩	🟩	🟩	🟩	|	🟩	🟩	
NFR2.1	🟩	🟩									🟩	🟩			|			
NFR2.2	🟩	🟩									🟩	🟩			|			
NFR2.3	🟩	🟩									🟩	🟩		🟩	|			
NFR2.4	🟩	🟩								🟩	🟩	🟩	🟩		|			
NFR2.5	🟩	🟩								🟩	🟩	🟩	🟩		|			
NFR2.6	🟩	🟩								🟩	🟩	🟩	🟩		|			
NFR2.7	🟩	🟩						🟩	🟩		🟩	🟩	🟩	🟩	|	🟩	🟩	🟩
NFR2.8	🟩	🟩									🟩	🟩	🟩		|			
FR3.0	🟩	🟩	🟩	🟦	🟦	🟦	🟦	🟩	🟩		🟩	🟩			|			
FR3.1	🟩	🟩	🟩					🟩	🟩		🟩	🟩	🟩	🟩	|	🟩	🟩	
FR3.2	🟩	🟩						🟩	🟩		🟩	🟩		🟩	|	🟩	🟩	🟩
FR3.3	🟩	🟩						🟩	🟩	🟩	🟩	🟩	🟩		|			
NFR3.0	🟩	🟩									🟩	🟩		🟩	|			
NFR3.1	🟩	🟩									🟩	🟩	🟩		|			
NFR3.2	🟩	🟩									🟩	🟩		🟩	|			
NFR3.3	🟩	🟩									🟩	🟩			|			

Figure 2: Requirement traceability matrix.

6 Initial Snapshots

Figure 3 is a screenshot of the main map screen after a room has been selected.

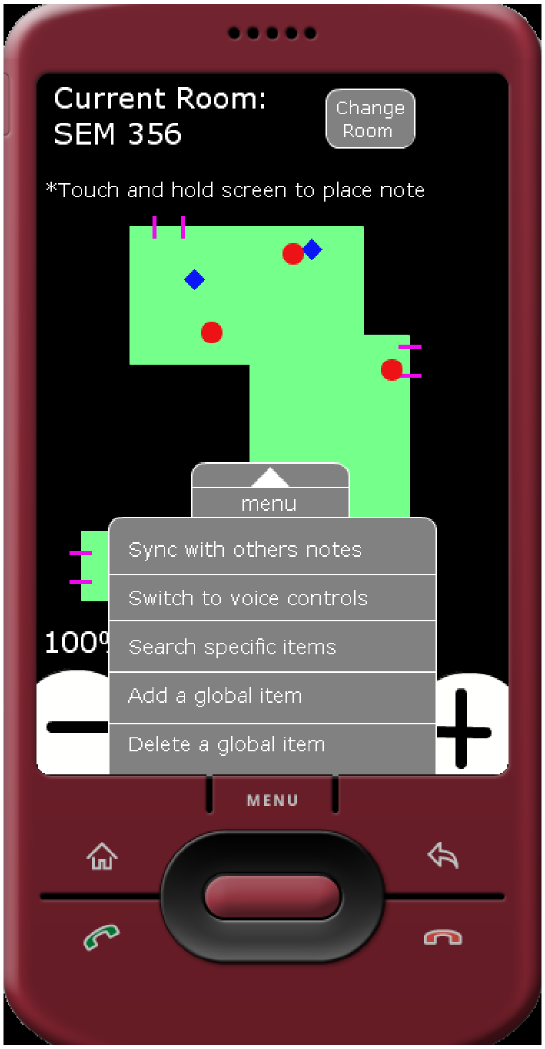
Figure 4 is a screenshot of the menu after the user presses the menu button.



Current Room:
SEM 356

*Touch and hold screen to place note

Figure 3: Current room selection.



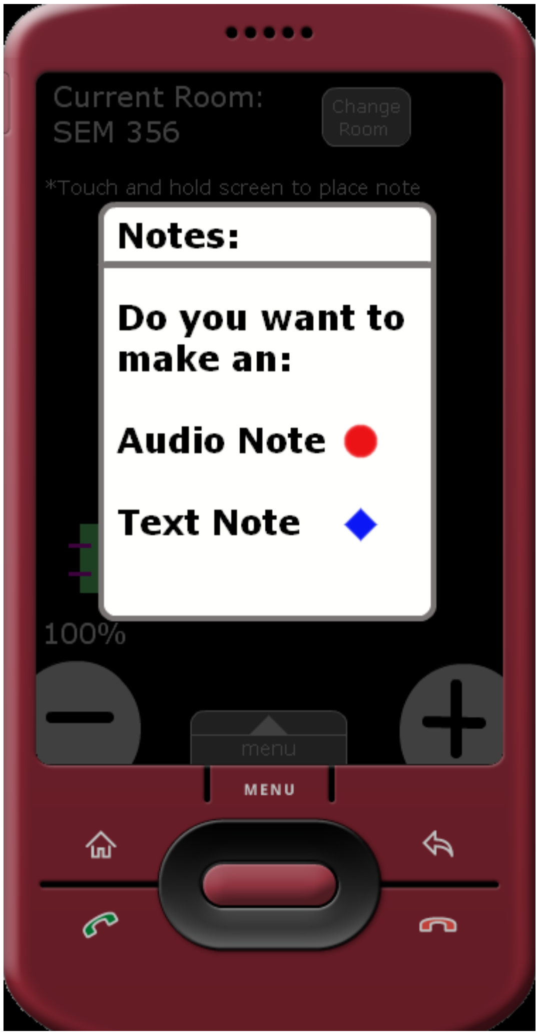
Current Room:
SEM 356

*Touch and hold screen to place note

Figure 4: Menu.

Figure 5 is a screenshot of the dialog box for making an audio or textual note.

Figure 6 is a screenshot of an audio note being recorded.



Current Room:
SEM 356

*Touch and hold screen to place note

Notes:

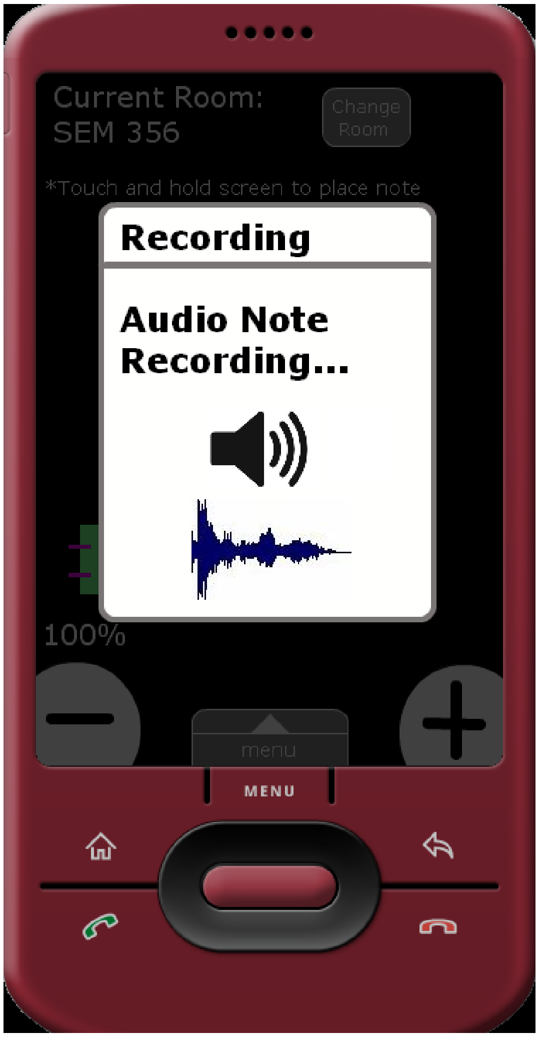
Do you want to make an:

Audio Note ●

Text Note ◆

100%

Figure 5: Note dialog.



Current Room:
SEM 356

*Touch and hold screen to place note

Recording

Audio Note Recording...

100%

Figure 6: Recording audio note.

Figure 7 is a photo of a sighted user using our application on the projected device.

Figure 8 is a photo of a blind user using our application on the projected device.

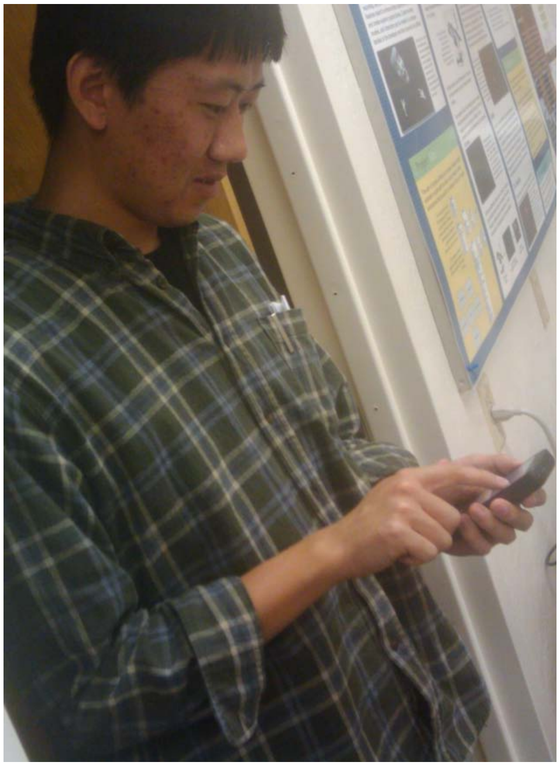


Figure 7: Sighted user.

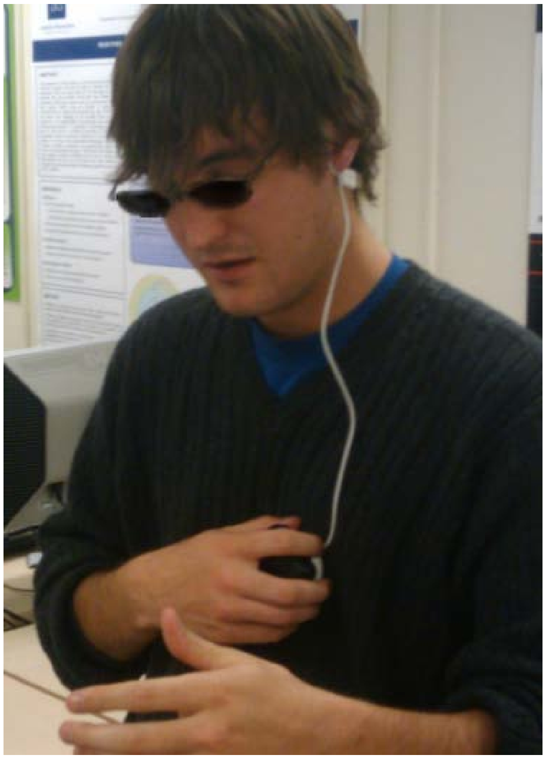


Figure 8: Blind user.

7 Glossary of Terms

Word	Definition
Accelerometer	A device that measures acceleration in a specific direction.
Android	A mobile operating system developed by Google based on Linux.
Audibly/Audible	The ability to make sound to be heard.
Birds Eye View	A literal top-down perspective view as if the viewer was a bird looking down on the land.
Blind	A person who cannot see.
Crowdsourcing	The act of outsourcing work to a crowd with interest in contributing to the work.
Database	A centralized location of data accessibly through the Internet.

Demographic	Selection of people who match a specified criteria.
Eclipse	The integrated development kit used to program and compile the source code.
Google	Famous search engine. Also the company who developed Android.
GPS	Global Positioning Satellite.
Gyrometer	A device used to measure the rotational speed of an object that is rotating.
Java	The programming language that is used for development on Android.
KML Model	The data format of the maps. It is based on XML.
Map	The drawing of the rooms and buildings.
Overlay	An image or part of an image composited over another image.
Room Item	Any object that is in the room such as furniture.
Sighted	A person that can see (not blind).
Textual	Based on text (words).
Text-To-Speech	The conversion of text to a sound.
Visually-Impaired	Person who is blind or has a vision handicap.

8 List of References

1. In Caverly's article he discusses how Google is experimenting with letting users submit location data (such as gas station and coffee shop locations combined with GPS coordinates) through Google Maps. Google then tries to filter the data to make sure it is good.

Caverly, Doug. "Google Maps Makes Use Of Crowdsourcing." 2007. WebProNews. 2009 <http://www.webpronews.com/topnews/2007/08/03/google-maps-makes-use-of-crowdsourcing>.

2. Harrington discusses how to map objects inside a building (such as cubicles, offices, conference rooms, etc.) using a plug-in for the image editing software GIMP. It can also graph the location of individuals or groups of employees based on job function, or track unused office space visually. However, this is not mobile and does not utilize crowdsourcing.

   Harrington, Nathan. "Map places, people, and relationships inside a building with open source software." 2007. IBM. 2009 <http://www.ibm.com/developerworks/library/os-gimpmap/>.

3. The article is a discussion piece regarding the difficulty a blind person has navigating a building for the first time. The author proposes a solution of a small, wearable device that emits audio or tactile feedback when near a Braille sign because the locations of Braille signs are not standardized. There are responses from blind patrons asking for what they feel is needed.

   "Navigating a Building for the First Time?" 2008. American Foundation for the Blind. 2009 <http://www.afb.org/message_board_replies2.asp?TopicID=4064&FolderID=3>.

4. Micello is an application for the iPhone that provides accurate and useful indoor maps of specific buildings. The program allows mapping of popular places such as shopping malls, casinos, stadiums, and other places people gather. This sort of indoor navigation with detailed information is what we are trying to create.

   "The Mind of a Building in the Palm of your Hand." 2009. Micello. 2009 <http://www.micello.com/>.

5. Waze was introduced to us by Dr. Folmer as a reference for how crowdsourcing works. Waze is a social mobile application which enables drivers to build and use real-time road intelligence. The information for the map is provided by crowdsourcing; constantly updated by people to provide the most accurate up-to-date info. This is what we want—except for inside buildings, not street maps.

   "Real-time maps and traffic information based on the wisdom of the crowd." 2009. Waze. 2009 <http://www.waze.com/>.

6. The book Semistructured Database Design discusses how to properly format and store data for web applications. The authors discuss the importance of normalization and efficiency using algorithms and XML.

   Dobbie, Gillian, Mong Li Lee, and Tok Wang Ling. Semistructured Database Design. New York: Springer, 2005.

7. Surowiecki discusses the psychology of crowd mentality and how as a group they are smarter than the smartest single member of the group. He also discusses how as a group, they can also be smarter than proclaimed experts in a subject due to the diversity and opinions. The subject matter relates to how we should use crowdsourcing for the project.

   Surowiecki, James. The Wisdom of Crowds. New York: Doubleday, 2004.

9 List of Differences from SRS 2009

The main differences with the project in its current form is its move from being a partially Navitar-based system to a system that is completely independent of Navitar. This decision was made because the two software systems are now being developed completely independently, and it doesn't make sense to try and integrate them along every step of the process.

In light of this different development path, we revaluated our priorities in the program and are now focusing more on a searching and finding algorithm for objects and their generalizability rather than navigating around them. What this means for our project is less emphasis on the actual interface for the user, and more on the core functionality of what really makes this a worthwhile project. Its value comes in being able to properly elicit and explain an accurate description of your surroundings and to respond to queries about the immediate surroundings.

Previously, we were also concerned with the data collection because we were worried about the filtering of the data that would need to come into the system. This is no longer the case as we are not concerning ourselves with the filtering at this time, since we are assuming that users will police themselves. Instead, the time we were going to spend developing this aspect is now being spent in making the user uploading process more streamlined and user friendly, for sighted and non-sighted users alike.

10 Contributions of Team Members

Kevin Grant:

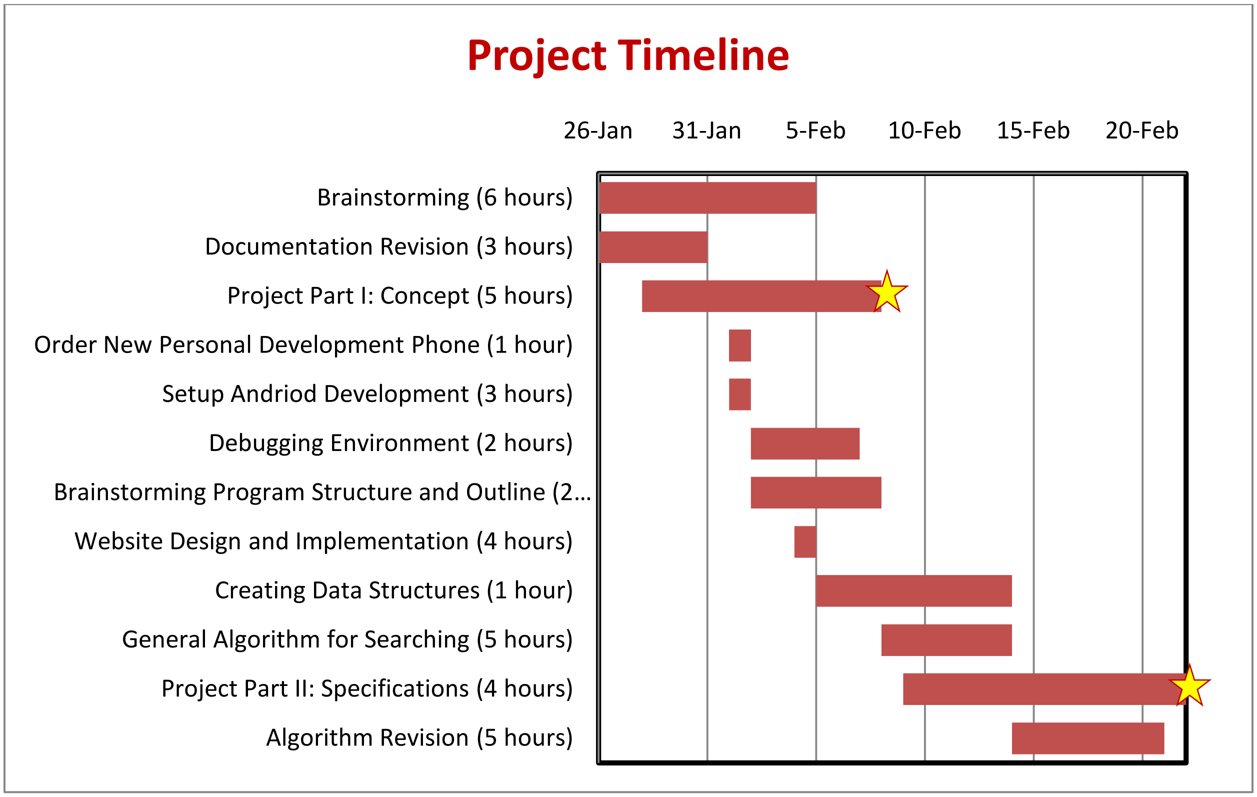
Primary focus is project management and primary programmer for the program's main architecture.

Angela Proffitt:

Primary focus is handling the paperwork for the project, making sure each paper is turned in on time, assisting in programming, and developing the voice command parser.

Alex Tam:

Primary focus is creating the graphical user interface and room map parser. Head Webmaster.


Project Timeline
26-Jan
31-Jan
5-Feb
10-Feb
15-Feb
20-Feb
Brainstorming (6 hours)
Documentation Revision (3 hours)
Project Part I: Concept (5 hours)
Order New Personal Development Phone (1 hour)
Setup Andriod Development (3 hours)
Debugging Environment (2 hours)
Brainstorming Program Structure and Outline (2...
Website Design and Implementation (4 hours)
Creating Data Structures (1 hour)
General Algorithm for Searching (5 hours)
Project Part II: Specifications (4 hours)
Algorithm Revision (5 hours)