Senior Reminder System Project

World Requirements Specification

SE 4351 – Requirements Engineering, Section 001
September 28, Fall 2015

Milton Bland, mxb120730@utdallas.edu
Zachary Calman, zxc120530@utdallas.edu
Ridge Frederick, rmf140130@utdallas.edu
Grant Freeman, gcf140030@utdallas.edu
Brad Gracy, bgj130030@utdallas.edu
Jeanie Handler, jch130930@utdallas.edu
Maria Haney, maria.haney@utdallas.edu
Justin Keeling, jxk120130@utdallas.edu
Mazen Lawand, mlj130430@utdallas.edu
Maryellen Oltman, mco130030@utdallas.edu
Kevin Szwagiel, kxs124230@utdallas.edu
Andrew Vaccaro, andrew.vaccaro@utdallas.edu
Dalton Wooley, dbw081000@utdallas.edu
Phillip Yellott, pay130030@utdallas.edu

Team Website: http://www.utdallas.edu/~atv130330
# Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>September 9, 2015</td>
<td>Initial Version.</td>
<td>Milton Bland, Zachary Calman, Ridge Frederick, Grant Freeman, Brad Gracy, Jeanie Handler, Maria Haney, Justin Keeling, Mazen Lawand, Maryellen Oltman, Kevin Szwagiel, Andrew Vaccaro, Dalton Wooley, Phillip Yellot</td>
</tr>
<tr>
<td>1.2</td>
<td>September 28, 2015</td>
<td>Copied 1.1 version and cleaned up for submission.</td>
<td>Dalton Wooley</td>
</tr>
</tbody>
</table>
Process

1. Project organization

1.1 Process model
This project shall use the Waterfall model as its process model, except it shall execute in stages. These stages are associated with the two phases of the project, and each phase shall have its own waterfall process. We have chosen the Waterfall model because we know the requirements shall not change in the middle of the phase so it shall be the most efficient process.

1.2 Organizational structure
Our team shall consist of three-four teams with a total of 14 members who shall work in tandem

**Documentation Team**
*Milton Bland* - Documentation Manager
*Jeanie Handler* - Documentation Manager
*Dalton Wooley* - Co-Project Manager, Documentation Manager

**Development Team 1**
*Ridge Frederick* - Developer
*Grant Freeman* - Co-Project Manager, Developer
*Brad Gracy* - Developer
*Maryellen Oltman* - Developer

**Development Team 2**
*Justin Keeling* - Co-Project Manager, Developer
*Kevin Szwagiel* - Developer
*Andrew Vaccaro* - Webmaster, Developer
*Phillip Yellot* - Developer

**Development Team 3**
*Zachary Calman* - Developer, Testing Developer
*Maria Haney* - Developer, Testing Developer
*Mazen Lawand* - Co-Project Manager, Developer

1.3 Organizational boundaries and interfaces
Due to the small team size, one team member shall be delegated to each task. The unused team member shall review and make revisions where necessary.

1.4 Project responsibilities
Both Co-Project Managers shall be involved with all phases of both portions of the project.
Co-Project Manager (CPM)
The CPM shall be responsible for managing the deliverable schedule and ensuring completion by due date.

Documentation Manager (DM)
The DM shall be responsible for all documentation and documentation control.

Developer
The Developer shall be responsible for the entirety of design and implementation of the software system. The developer shall also be responsible for meeting all scheduling as designated by the CPM.

Webmaster
The Webmaster shall be responsible for uploading and managing deliverables onto the team website. The deliverables shall be ready on the team website by the due date.

Testing Developer
The Testing Developer shall work with the DM and CPM to create, document, and save test cases and their results.

2. Managerial process

2.1 Management objectives and priorities
Management, which comprises of the Co-Project Managers, is responsible for getting activities completed efficiently and effectively. The main objective of the management is to organize the meetings for discussions, check the status of the project, and submit the project on time.
The main objectives are:
- Scheduling
- Directing
- Reviewing
- Submitting Deliverables

2.2 Assumptions, dependencies, and constraints

Assumptions
- Any difficulty with an assigned task is to be communicated to the other team member
- The professor/end customer shall not make changes in the requirements or scope
- Professor/end customer shall clarify any doubts, concerns, or uncertainties

Dependencies
- Each CPM is reliant upon the developer to finish their duties on schedule
- Each developer is reliant upon the CPM to clearly delegate their tasks and their schedule
- The Webmaster is reliant on the DM to provide the deliverables to upload to the team website
Constraints

- The time frame of 6 weeks for the project shall make scheduling essential to project completion
- The time frame of 6 weeks for the project shall make understanding of the Android SDK essential to task completion to schedule
- The quality of the project is dependent upon the requirements of the project

2.3 Risk management

Risks and Contingency Plans

- One of the team members is unable to complete his task on schedule
  - The team member shall be responsible for communicating this issue to the other team member so they can complete the task on schedule.
- Loss of project data or progression
  - Project documents shall be hosted in the cloud. Project source code shall be version controlled using Git and hosted on GitHub.
- Lack of experience or skill in a required area
  - The team members shall be responsible for researching the skill using resources such as the Internet, the professor, on-campus resources.
- Poor quality
  - Each team member shall review the other team member’s work to ensure that it fulfills quality requirements.
- Change in requirements
  - Team members shall adjust the schedule and requirements to meet any changes.
- Failure to complete project
  - Team members shall adhere to the schedule and complete tasks on time to ensure the project is completed.

2.4 Monitoring and controlling mechanisms

Each delivery phase shall be lead by both Co-Project Managers. Project documents and deliverables shall be controlled with online cloud hosting and version control such as Google Drive and GitHub.

2.5 Meetings

On a weekly basis, each CPM will review and meet with their team as necessary. If nothing is scheduled to be completed before the next meeting, then the meeting is not required. Meetings will be scheduled over SMS for all team members who can attend.
3. Technical process

3.1 Methods, tools, and techniques

**Tools**
The following tools shall be used for development of documentation and code:
- **Java**: The development language for program implementation
- **Google Drive**: The document editor and cloud host for documentation
- **Android Studio**: The IDE used for writing source code and porting to android system
- **Git**: The version control system for source code
- **GitHub**: The cloud host for our git repository
- **SMS**: Our main form of communication

**Techniques**
Software development shall follow standard Java naming convention and Object-Oriented structure.

3.2 Software documentation
The following documentation shall be written:
- **Project Plan**
- **World Requirements Specification**
- **Program Specification**
- **User Manual**

**Phase I**
- **World Requirements Specification**
- **Program Specification**
- **User Manual**

**Phase II**
- **World Requirements Specification**
- **Program Specification**
- **User Manual**

3.3 Project support functions
- **Version Control**
  - Source code shall be version controlled with Git and GitHub.
- **Quality**
  - The GitHub issue tracker shall be used to keep track of issues and tasks that need to be completed.
- **Documentation**
  - Google Drive shall be used to write all documentation.
Introduction

Purpose
The purpose of this document is to compile all documentation on the SE 4351 Preliminary Project Phase II, or the Senior Reminder System Project. This documentation will include the Requirements Specification containing the functional and nonfunctional requirements for this project, the Program Specification containing the implementation of the project, and the User Manual. Having all of this documentation will outline for the reader the motivations and decisions that shaped the development of this project.

Scope
The project is defined by the boundaries of the selection process and our narrowing of the project definition. Project selection was completed by meeting together and each person suggesting an idea. After discussion of the upsides and downsides of the various ideas, our team came to a unified agreement on a reminder system. Review of the project goals, deliverables, tasks to complete, their associated costs and deadlines, further narrowed our project to a precise application idea.

Objectives and Success Criteria
The goal of this project is to create an Android App to help elderly people manage events and important tasks like taking their medicine. In order to better define the behavior of this system, we devised a Requirements Specification further clarifying what functionality this system is required to fulfill, such as front-end functionality and back-end constraints.

Definitions
SR - Senior Reminder
The formal title of our phase 1 project abbreviated for ease of writing and reading.

Android
Mobile Operating System developed by Google Inc.

Object-Oriented
A methodology that enables a system to be modeled as a set of objects that can be controlled and manipulated in a modular manner.

App
A program or piece of software designed and written to fulfill a particular purpose of the user.

Category
A category is a descriptor containing the multi-dimensional vocabulary items having a similar meaning, relation and/or purpose.
Disjoint Category
A disjoint category is one that does not have its items overlap with any other category.

Overlapping Category
An overlapping category is one that has one or more of its items overlap with items in other categories. Categories can be either activity-based or item-based at the root level e.g. items as in ‘Food’, ‘Drink’, ‘People’ etc. and activities like ‘I want to eat’, ‘I want to go’ etc.

Preliminary Domain

Overview
Our team is to architect and implement an Android. The goal of this system is to allow management of notifications and reminders for the visually impaired and those with memory-loss.

PD 1. Assist the elderly with problems of vision loss, memory loss, and muscle weakness with daily living functions.
PD 2. Utilize visual aids to communicate with elderly users with vision impairment and memory loss.
PD 3. Use a sorting of categories of actions and items to assist the user in selection of tasks.
PD 4. Handle emergency situations with efficiency and quickness.
PD 5. Provide a usable search option to easily navigate the system.

Issues with Preliminary Definition

PD 1. Assist the elderly with problems of vision loss, memory loss, and muscle weakness with daily living functions.
Issue 1: Definition scope is too broad. Needs to be focused to a smaller subset of the problem.
Issue 2: Apps are most useful when handling a well defined problem as opposed to a broad one such as elderly health issues.

PD 2. Utilize visual aids to communicate with elderly users with vision impairment and memory loss.
Issue 1: Definition scope is too broad. Needs to be focused to a smaller subset of the problem.
Issue 2: Time to implement is not sufficient to address all forms of vision impairment and memory loss.

PD 3. Use a sorting of categories of actions and items to assist the user in selection of tasks.
Issue 1: It will be difficult to define categories that are meaningful and unique (due to overlapping categories).
Issue 2: Creating new categories might lead to user difficulty in allocating tiles.

PD 4. Handle emergency situations with efficiency and quickness.
Issue 1: It will be difficult to define what is an actual emergency
issue 2: It will be difficult to detect when an emergency happens

PD 5. Provide a usable search option to easily navigate the system.
Issue 1: Search function will not be a useful addition to the app.
Issue 2: Voice Search will consume an excessive amount of resources that may be detrimental to the overall.
Issue 3: Voice analysis and detection will have to be performed by an outside service because that single function alone exceeds the resources of our team.

Functional Requirements
1. The SR system shall display a visual tile interface for interacting with reminders.
   1.1. The SR system shall allow the user to customize reminders to make them more meaningful.
      1.1.1. The SR system shall allow image association to individual reminders.
      1.1.2. The SR system shall allow editing of the tile color theme.
   1.2. The SR system shall create reminders.
      1.2.1. The SR system shall fill in reminder information through voice-to-text.
   1.3. The SR system shall update reminders.
      1.3.1. The SR system shall allow modification of the reminder information.
   1.4. The SR system shall delete reminders.
      1.4.1. The SR system shall ask for confirmation before deletion.
   1.5. The SR system shall retrieve reminders for viewing.
      1.5.1. The SR system shall
2. The SR system shall require an emergency contact phone number.
   2.1. The SR system shall contact this contact through SMS in the case of missed reminders.
3. The SR system shall notify the user through the Android system about reminder events.
   3.1.1. The SR system shall push notifications at requested times.
   3.1.2. The SR system shall play alarms at requested times.
   3.1.3. The SR system shall notify emergency services in emergency situations.
      3.1.3.1. The SR system shall have a quick access key to 911 services.
4. The SR system shall allow management of the data.

Non-Functional Requirements
1. The SR system shall be easily understood.
   1.1. The SR system shall be well commented.
2. The SR system shall be portable.
   2.1. The SR system shall be able to run on any system that supports Android API 15/Version 4.0.3.
3. The SR system shall be enhanceable.
   3.1. The SR system shall facilitate future additions to the source code
       3.1.1. The SR system shall have a simple build system.
       3.1.2. The SR system shall be open sourced.
3.2. The SR system shall use an Object Oriented Architecture.
4. The SR system shall be reusable and adaptable.
   4.1.
5. The SR system shall have good performance and be responsive.
   5.1. The SR system shall be fast.
       5.1.1. The SR system shall complete its tasks in less than 0.5 seconds.
   5.2. The SR system shall minimize memory usage
       5.2.1. The SR system shall avoid making unnecessary copies of data in memory.
6. The SR system shall be user-friendly.
   6.1. The SR system shall be error-free
       6.1.1. The SR system shall complete executing without any errors.
   6.2. The SR system shall be well documented
       6.2.1. The SR system shall have a User Manual that explains how to use the system.
7. The SR system shall use simplistic interfaces.
   7.1. The SR system shall use visible high contrast icons.
       7.1.1. The SR system shall only display icons that are in contrast to the user interface placed behind it.
   7.2. The SR system shall use tiles with consistent coloring and images to visually organize.
       7.2.1. The SR system shall have a coloring scheme.
       7.2.2. The SR system shall have a set of distinct images associated with each type of reminder.

Traceability

Program Specification
The program specification for this project is available in the src/ directory of the deliverable .zip this document came in, or from our project website: http://www.utdallas.edu/~atv130330
User Manual

Appendix

Test Cases

Test cases not yet developed.

References


