Project Management Document

Smart Dialer Application

CSC 4330

Prepared by Cameron Moore, Omar Qasem,
Patrick Richardson, Matthew Spedale, and Jeffrey Tolliver

Submission Date: April 30th, 2017

Table of Contents

Index

Vision

The Smart Dialer application is a telephone-calling application designed to run on Android mobile devices running 5.0 Lollipop operating system or higher. The product is intended to simplify mobile phone calls, and provide more information to the application user upon receiving or making a call than what is provided by the default caller application that comes preinstalled on Android devices.

The main feature of the application is an automated reverse phone number (411) lookup. When the user makes a phone call, Smart Dialer stores the number inputted, performs a reverse phone number lookup on it for relevant information (company name, location, thumbnail image, and email) and stores it in the history list. Additionally, features such as a contact list, call logs, and speed dial are also integrated into Smart Dialer to ensure that users do not need to use both Smart Dialer and the default caller application depending on what functions they want to use.

Our strategy is to release this app on the Google Play Store as an alternative to the existing phone dialing application installed on Android products, to offer Smart Dialer as a more powerful and user-friendly application.

Configuration Management Plan

For our project, our group used the website GitHub, a web-based version of Git used for version control. Using GitHub was a requirement specified by Professor Nash, though our group planned to use it for version control independently. All of us had some experience with Git and GitHub from other Computer Science classes as well as independent programming, though utilizing GitHub for our Smart Dialer project has provided more experience with importance of well-planned version control.

Our main use of the GitHub repository was twofold. First, it served as a central storage space for all of our code, which was extremely helpful due to GitHub's version control and code conflict management tools. It also allowed each member of the team to be up to date on the latest version of our code, and ensure that no one was working on outdated versions. The second function GitHub provided was a storage space for all of our documentation. Most of the documentation was handled by team member Matthew Spedale, with additional polishing and feedback provided by the rest of the group. GitHub allowed us to create a folder separated from our actual code so that each member could access the documentation as needed, without having to worry about documentation potentially interfering with the application itself.

The code of Smart Dialer was all written using Android Studio's IDE, and the language we utilized was Java. The documentation was written using Microsoft Word, using the standard format seen in most documentation we viewed online and from Professor Nash's presentations.

Process Model

To ensure continuous progress was made on Smart Dialer, our team utilized the Agile Methods software development methodology. Agile's emphasis on flexibility and continuous refactoring was important to our team, as we wanted to be able to change our application as needed on the fly. We heavily relied on many of the twelve facets of extreme programming (XP) seen on Professor Nash's "Software Process" Powerpoint. Specifically, we often used simple design, pair programming, refactoring, and continuous integration. Pair programming in particular was extremely valuable to us, as there was a large discrepancy in Android Studio and Android development experience among team members. Our team met at least once a week, sometimes more often, and we would usually code together as a group. Pair programming allowed us to solve problems together and ensure all members had one uniform vision for the application.

Deliverables

1) Application:

For the initial release of the Smart Dialer application that culminates with the end of Professor Nash's Software Engineering class, The Smart Dialer executable is designed to only run on the Android mobile platform, specifically on Android smartphones using the 5.0 Lollipop operating system or higher. The main reason for this is simplicity, as the limited time period of the class would make programming the application for both Android and iOS difficult. Additionally, developers can publish applications to Android's Google Play Store for free, whereas on iOS there are costs to publish to the App Store. In the future, we may port Smart Dialer to iOS, but this would be outside of the scope of Professor Nash's class.

2) Documentation:

a) System Requirements Specification (SRS) Document

Smart Dialer's System Requirements Specification Document outlines the functional and nonfunctional requirements of Smart Dialer as precisely as possible, as well as a detailed outline of our software application. It describes the user interface, hardware requirements, functionality descriptions, and more. It also describes the intended client and audience of our product. The document was revised over the class duration in order to accurately reflect changes made to Smart Dialer.

b) Software Design Document (SDD)

Smart Dialer's Software Design Document outlines the implementation for the intended form of the application, including the main system and sub-system architectures, the logical, development, physical, data, and work assignment views, and the user interface of the project and application.

c) Software Testing Document (STD)

Smart Dialer's Software Testing Document

d) Smart Dialer Code

Potential Risks

The potential risks associated with the development of Smart Dialer primarily concerns the appropriate management of time. Since the application was created for Professor Nash's Software Engineering class, the application needed to be fully developed in time for the end of the Spring semester, when the finalized applications are due for the final in-class presentation. Additional risks include the usage of Android Studio for development and the team dynamics. The first point involves that only two of our five members had significant experience programming using Android Studio. This could prove to be a problem if learning to utilize Android Studio fully in addition to all of our other class work proved to be too difficult. Our team minimized this risk by utilizing a lot of group and pair programming, where multiple people would be working on the application code using one computer. The second point is that our team dynamic could prove to be incompatible, and cause trouble to the overall development. Our team members all knew each other before joining as a team, and most of us had even worked on various projects together before, which largely mitigated incompatibility being an issue.

Team Members

1) Team Organization

Our team utilized the egoless approach for our development. We decided this was best for the project because due to the nature of the application being developed for a class in which our individual contributions to the project were part of the overall grade, each member of the team needed to have quantifiable contributions to the application. Though Patrick Richardson was our de facto team leader due to his experience with Android Studio and mobile application development, our team largely worked within the egoless approach, with Patrick’s leader role serving mostly to give us a singular direction in regards to development.

2) Team Member Contributions

a) Cameron Moore

-

b) Omar Qasem

-

c) Patrick Richardson

-

d) Matthew Spedale

-

e) Jeffrey Tolliver

-

Project Schedule

1. PERT Estimation Method
2. Critical Path Method
3. Gantt Chart

Meetings Summary

The following section of this document outlines all team meetings held during the period of this class.

3/16/2017:
Time: 1:30PM – 4:30PM
Attendance: Cameron Moore, Omar Qasem, Patrick Richardson, Matthew Spedale, and Jeffrey Tolliver
Objectives:

- Complete SRS document
- Get all group members connected to the project GitHub repository
- Have all group members install Android Studio

Outcome:

- SRS document largely completed with some final revisions
- All members successfully connect to the project Github repository
- All members successfully installed Android Studio

3/30/2017
Time: 3:00PM – 6:00PM
Attendance: Cameron Moore, Omar Qasem, Patrick Richardson, Matthew Spedale, and Jeffrey Tolliver
Objectives:

- Allocate programming and design tasks to each group member
- Familiarize group members with Android Studio
- Create new GitHub repository to fix Git issues.

Outcome:

- Each member given a definitive role in the group
- Additional experience with Android Studio
- Git issues resolved, new GitHub repository created

3/31/2017
Time: 1:30PM –5:00PM
Attendance: Cameron Moore, Omar Qasem, Patrick Richardson, Matthew Spedale, and Jeffrey Tolliver
Objectives:

- Add .gitignore file to repository
- Clean repository of ignored files
- Have each member create their own branch repositories
- Development of Log, Contacts, Dialer pages in Smart Dialer application

Outcome:

- Added .gitignore file to repository to successfully ignore unneeded files
- Cleaned repository of files .gitignore usually omits
- Individual branches created
- Progress made on development of Log, Contacts, and Dialer pages in Android Studio

4/7/2017
Time: 3:00PM – 4:30PM
Attendance: Omar Qasem, Patrick Richardson, Matthew Spedale, and Jeffrey Tolliver
Objectives:

- Re-implemented the app layout to best fit the design's needs
- Planned spring break meetings and activities

Outcome:

- Successfully altered app layout
- Defined goals to be completed over spring break

4/11/2017
Time: 2:00PM – 9:00PM
Attendance: Cameron Moore, Omar Qasem, Patrick Richardson, Matthew Spedale, and Jeffrey Tolliver
Objectives:

- Collaborative group work on application

Outcome:

- More progress made on development of application, refinement and added features to each page
- Individual work completed by each member (programming, documentation, and GitHub management)

Terms and Conditions