**MINISTRY OF EDUCATION AND TRAINING**



MOBILE PROJECT MANAGEMENT

MPM

**CAPSTONE PROJECT**

Specialty: Software Engineering

Project members:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Supervisor: Mr. Cao Xuan Vinh

Hanoi – 2013

RECORD OF CHANGES

\*A – Added M – Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Effective Date** | **Author** | **A/M/D** | **Change Description** | **Version** |
| 24/05/2013 | KhanhVD | A | Newly Created | 0.1 |
| 30/05/2013 | KhanhVD | A | Add Report No.1 | 0.2 |
| 04/06/2013 | KhanhVD | A | Add Report No.2 |  |
| 04/06/2013 | KhanhVD | A | Add Report No.3 |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |

**Table 0-1: Record of Changes**

[**A.** **INTRODUCTION** 4](#_Toc359507342)

[**I.** **Project Management System Introduction & History** 4](#_Toc359507343)

[**II.** **Initial Ideal of the Group** 4](#_Toc359507344)

[**III.** **Existing Products& Systems** 5](#_Toc359507345)

[**1.** **2Do: To Do List | Task List** 5](#_Toc359507346)

[**2.** **GTasks: To Do List & Task List** 6](#_Toc359507347)

[**3.** **Air To-Do** 6](#_Toc359507348)

[**4.** **Comparison with our product** 7](#_Toc359507349)

[**IV.** **Our solution and Purposes** 8](#_Toc359507350)

[**1.** **Solution and Improvement** 8](#_Toc359507351)

[**a.** **Mobile Application Development Overview** 8](#_Toc359507352)

[**b.** **Android Platform** 8](#_Toc359507353)

[**c.** **Cloud Computing** 9](#_Toc359507354)

[**2.** **Our purposes** 9](#_Toc359507355)

[**B.** **PROJECT MANAGEMENT PLAN** 9](#_Toc359507356)

[**I.** **Project Overview** 10](#_Toc359507357)

[**1.** **Project Name** 10](#_Toc359507358)

[**2.** **Project Objectives** 10](#_Toc359507359)

[**3.** **Project Scope** 10](#_Toc359507360)

[**II.** **Project Environment** 10](#_Toc359507361)

[**1.** **Production Environment** 10](#_Toc359507362)

[**2.** **Development Environment** 11](#_Toc359507363)

[**3.** **Tools and Techniques** 11](#_Toc359507364)

[**III.** **Project Organization** 12](#_Toc359507365)

[1. Software Process Model 12](#_Toc359507366)

[2. Roles and Responsibilities 14](#_Toc359507367)

[**IV.** **Project Management Plan** 14](#_Toc359507368)

[1. Tasks 14](#_Toc359507369)

[2. Task Sheet: Assignment and Timetable 17](#_Toc359507370)

[**V.** **Coding Convention** 17](#_Toc359507371)

[**C.** **SOFTWARE REQUIREMENT SPECIFICATION** 21](#_Toc359507372)

[**I.** **Introduction** 21](#_Toc359507373)

[**1.** **Objectives** 21](#_Toc359507374)

[**2.** **Scope** 21](#_Toc359507375)

[**II.** **Requirements** 21](#_Toc359507376)

[**1.** **User Requirement Specification** 21](#_Toc359507377)

[**4.** **External Interface Requirement** 25](#_Toc359507378)

[**a.** **User Interfaces** 25](#_Toc359507379)

[**b.** **Hardware Interfaces** 25](#_Toc359507380)

[**c.** **Software Interfaces** 25](#_Toc359507381)

[**d.** **Communication Protocol** 25](#_Toc359507382)

[**5.** **Non-Functional Requirement** 26](#_Toc359507383)

[**a.** **Performance** 26](#_Toc359507384)

[**b.** **Usability** 26](#_Toc359507385)

[**c.** **Scalability** 26](#_Toc359507386)

[**d.** **Maintainability** 26](#_Toc359507387)

[**e.** **Availability** 26](#_Toc359507388)

[**f.** **Security** 26](#_Toc359507389)

[**6.** **Functional Requirement** 26](#_Toc359507390)

[**a.** **List of Functions** 26](#_Toc359507391)

[**b.** **Function Description** 27](#_Toc359507392)

[**D.** **DESIGN & IMPLEMENTATION** 68](#_Toc359507393)

[**E.** **TEST DOCUMENTATION** 68](#_Toc359507394)

[**F.** **USER’S MANUAL** 68](#_Toc359507395)

[**G.** **APPENDIX** 68](#_Toc359507396)

1. **INTRODUCTION**
2. **Project Management System Introduction & History**

In the modern lift, people must often face with pressures from work office, family and social. There is a long list of individual work, family work, agency work as well as social activities. All those tasks need to be remembered and carried out on time as well as exactly. Many people have their own way to remember and organize their tasks. Generally, the simplest way is using the notebook to take notes and plan their tasks. Before 1998 when Microsoft published windows 98 and mobile phone was expensive product, if you had asked somebody about how to organize the work, they would shown to you their small notebooks, or simply some piece of papers which recorded the daily tasks was stuck at the visible places that was easy to see, such as the fridge, calendar or desk office. In the first years of twenty century, personal computer became popular. People used to use Sticky Notes (the software that is built into Windows) or others software to manage their tasks. Furthermore, most of the first generations of cell phones of Nokia, Motorola and Black Berry have owned the remindful software that allow user to create simple schedule which was installed in their phone available.

Nowadays, Smartphone has gradually replaced traditional mobile phones. Therefore, a series of applications for task management and schedule programming have been dramatically developed with a lot of powerful functions to assist the task management and work organization by programmers. By using a few simple steps, users can build the list of tasks with the functions of notification, alarm, sorting and checklist. Nevertheless, most applications only support personal task management, not having those functions in group project tasks’ management. Besides, it has not been yet the functions of sharing job or communication among people within using the same system.

These are the reasons why people need an integrated system that allow users can manage tasks, task sharing and communication channels to support their works.

1. **Initial Ideal of the Group**

The project’s main objective is to build systems consist to build and manage tasks, combine share tasks and communication systems pass by message among users. Our purpose is that people can apply our application in project management.

After reading a research report on the Smartphone market in Vietnam, we realized that Smartphone using the operating system is concerned Android and dominate the market. We decided to write an application android

After discussing about trends in software development, we know that SOA (Service Oriented Architecture), Cloud Computing and Mobile Application are the favorite topics are being discussed on many forums technology

Our team makes final decision that our system will include:

* Android device is client system that is used for user to interact with the system. It can be also developed on other mobile platform such as iOS, Windows Phone, Windows Store.
* Web services are the backend of mobile project management system that will include business logic analyst and database processing. Our web service is Platform-Independent Model and will support developers can use our service for their application in the many development platform such as Web Form, mobile application, windows store application.
* By using cloud computing for deploying web service and database, we can use advantage of cloud computing such as high scalability, high performance, high security… In the other hand, it helps us save money to publish and deploy our application like a commercial product.

1. **Existing Products& Systems**

Currently there are several similar products

1. **2Do: To Do List | Task List**



**Figure A-III-1: 2Do: To Do List**

**2Do: To Do List** is a software product from **Guided Ways Technologies Ltd**. 2Do: To Do List is introduced to be able to run on 3 environments (Android, iOs, Mac OS) and support Smartphone, tablets, and Mac computer. 2Do is incredibly powerful software. This feature-packed productivity app provides everything you need to manage your daily tasks. 2Do includes a helpful tutorial on how to get started. Creating a new to-do list is pretty easy, and the app includes a variety of options for each task. For each item on your 2Do list, you can also add notes, set a due date, tag a location, schedule an alarm, and tack on a URL or photo. You can even make an audio recording with voice notes.

1. **GTasks: To Do List & Task List**

****

**Figure A-III-1: GTasks: To Do List & Task List**

**GTasks** is a software product from **Appest Inc.** Gtask is run only on Android. It supports smart phone and tablets. Gtasks is my favorite To-do list app, because it’s simple and efficient task management. In-app, you can set ring tone reminders, create individual tasks, or categories of lists. You can change the name, color, order of your lists and sync with your Google tasks perfectly. With the simple swipe of a finger, you can scroll between the categories, and quickly pull up what you’re supposed to be doing next. The widgets are good-looking, and flexible

1. **Air To-Do**

****

**Figure A-III-1: Air To-Do**

**Air To-Do** is a software product from AllAboutApps Inc. Air To-Do is run on iOS 3.1 or later. It is compatible with iPhone, iPod touch, and iPad. Air To-Do’s interface is very simple, which is a plus for a list app. For each individual item, you can add a due date, set an alert, add a URL or photo, write a note, or tag a location. Air To-Do also includes plenty of sharing options, including the ability to send items via email or text message. One thing I really like about the Air To-Do app is its integration with Facebook and Twitter.

1. **Comparison with our product**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Products**  **Features** | 2Do: To Do List | GTasks | Air To-Do | **(MPM)** |
| Add Tasks | X | X | X | **X** |
| Drag & Drop Task | X |  |  | **X** |
| Share task |  |  |  | **X** |
| Comment on task |  |  |  | **X** |
| Send message |  |  |  | **X** |

**Table A-III-1: Comparison with our product**

1. **Our solution and Purposes**
2. **Solution and Improvement**

We will develop Mobile Project Management by using cloud computing technology and Android platform for client side’s development. These are overview about our technical solution:

1. **Mobile Application Development Overview**

Today [mobile phone](http://www.whatech.com.au/index.php?option=com_cstatistic&task=tracklink&url=bG9jfDk2NTZ8aXQtcGFnZXMvc2VhcmNoL21vYmlsZStwaG9uZQ%3D%3D) has become an important gadget in human life. With some people, it is impossible to do many works without mobile phones. Many big, medium and small enterprises are running their many business activities with the help of different apps. Mobile phones are playing a significant role in business activities of many organizations through apps for marketing, sales, and other trading activities. According to a report about shopping trends on the internet, by 2015, the world have more than 1.3 billion people who will be using smartphone for their daily life.. The report forecasts that the mobile application market could be grow to $ 50 billion in the next two years..Mobile application development is big trend in IT solutions and there are more and more IT Companies will join into this field in this year and the number of students who are studying in mobile application development is increase very fast. BlackBerry, Android, Windows, iPhone, etc. are some of the most popular [mobile phone](http://www.bizrate.com/mobile-phone/index__af_assettype_id--4__af_creative_id--3__af_id--%5bAFF-ID%5d__af_placement_id--%5bAFF-PLACEMENT-ID%5d.html) applications that are available in the market. All of them are very much liked by people as they help in exploring different things at the same time. In our capstone project, we chose Android Platform for Mobile Application Development.

1. **Android Platform**

Android is an open-source software stack created for a wide array of devices with different form factors. The primary purpose of Android is to create an open software platform available for carriers, OEMs, and developers to make their innovative ideas a reality and to create a successful, real-world product that improves the mobile experience for end users. We also wanted to make sure that there was no central point of failure, where one industry player could restrict or control the innovations of any other. The result is a full, production-quality consumer product whose source is open for customization and porting

**Benefits of Android Platform**

* **Graphic Support**

Android offers high built-in support for power 2D and 3D graphics, which help businesses to attract maximum users to their mobile applications. High quality graphic plays a vital role in apps.

* **Cost Effective**

Android is very cost effective as an open source platform. The wide range of android development tools are free to download. Thus, mobile application development companies can deliver high quality apps at affordable rates to businesses that are always looking for solutions that are cost effective.

* **Freedom to Developers**

Android platform is quite flexible and therefore mobile app developers can work with greater flexibility and freedom. This open source technology gives freedom to developers to extend the source code and exercise their development capabilities and skills to create an effective and unique app.

1. **Cloud Computing**

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction

**Benefits of Cloud Computing**

* **Cost Savings**

Cloud computing providers (Amazon, Microsoft...) have pricing calculator method to help customer can estimate usage cost. With small business, pricing is about $500 per month.

* **Better Scalability and High Performance**

Cloud Computing give the opportunity for customers to scale their computing resources whenever they deem it by increasing or decreasing the required resources. You're not paying for resources which you are not utilizing. If your websites or services have too many connection in the same time, cloud computing will duplicate all website resources such as database, data. This feature helps the website always be available.

* **Easier Maintenance**
* **Better Support**

Cloud Computing providers (Microsoft, Amazon) have great support team and they always on the line for the helps. Moreover Cloud Computing supported multi programming language such as C#, Java, PHP, Python…It is easy to find sample or tutorial on Internet.

1. **Our purposes**

This project is registered and implemented as the capstone project for the team members. The first purpose is to fulfill the requirements from FPT University studying program. The second purpose is to create a complete product for going lives. Furthermore, we have strongly believed that our application can help people to manage their personal task and enterprise company to manage project. We will publish our application like commercial app as soon as we can.

1. **PROJECT MANAGEMENT PLAN**
2. **Project Overview**
3. **Project Name**

The name of this project is “Mobile Project Management”. It project the main purpose of this project is to helps organizations or companies to manage their projects or people to manage their personal tasks.

The project aims to officers who have trouble in organize their tasks. They often work on the many projects in the same time. So we need the application that helps to manage and organize tasks, can run on their personal device (such as mobile phone, PDA or tablet). Our application gives them solution to solve their problem.

1. **Project Objectives**

This mobile application helps project manager and team members manage schedule, tasks, communication and human resource in the project. We will develop this application by using WCF Services for Android device. Services and Database will be deployed on Microsoft Azure which is one of the largest cloud computing providers in the world. The application must be reliable, fast, friendly, and easy to use.

1. **Project Scope**

The scope of the system:

* The project manager in project management can use this app for building project and organization features in our system
* The team member is allowed create and share tasks for others members in the same organization.
* All members can share their comments and experience by using task comments and sending message feature.
* A person can find and invite his/her friends for working on a project.
* A person can manage his/her individual work.

Target users of the system:

* Medium and small companies are running their many business activities with the help from mobile application.
* Consumers who need the mobile app for individual tasks management.
* Students who word in small team for each school subjects.

1. **Project Environment**
2. **Production Environment**

|  |  |
| --- | --- |
| Minimum Configuration | |
| Platform Version | Android OS version 3.0 and more. |
| Chipset | Exynos 4412 |
| CPU | Quad-core 1.4 GHz |
| GPU | Mali-400 |
| RAM | 1 GB DDR3 |
| Internal Storage | 8GB Flash |
| External Storage | microSDHC Card |
| Screen size | 4.7" |
| Screen type | HD |
| Screen resolution | 480 x 800 pixels |
| Multi –Touch | Yes |
| Camera | Yes |
| Connection | Wireless and 3G |

1. **Development Environment**
2. **Hardware Requirements**

* Personal computers for developing/testing with the minimum configuration: 2GB of RAM, 100 GB of hard disk, Core 2 Duo 2.0 GHz.
* A server computer for run developing/testing WCF services on Window Azure Computer Emulator with the minimum configuration: 4GB of RAM, 100GB of hard disk, Core 2 Duo 2.0 GHz.
* Android Smartphone with 3G and Wi-Fi powered for testing and deploying purposes

1. **Software Requirements**

* Operating System: Windows 7, Windows 8
* Framework: .NET Framework 4.5, ASP .NET Web Services
* IDE: Visual Studio Express 2012 for Web, Eclipse Juno 4.2.2 with ADT Plugin
* Others: MS Office, MS Project, Adobe Photoshop CS5
* An account of Microsoft Azure(Educator) for deploying Services and Database.
* GitHub for Windows Version 1.0.48.0

1. **Tools and Techniques**
2. **Tools**

|  |  |  |
| --- | --- | --- |
| **Tools** | **Description** | **Version** |
| Microsoft Office Project | Project management software, support for planning, assigning task, control progress… | 2007 |
| Microsoft Office | Use Word, Excel, PowerPoint, Visio to make reports, create charts, draw diagram and make presentations | 2010 |
| Visual Studio Express for Web | WCF Service development purpose | 2012 |
| Eclipse | Android application development purpose | 4.2.2 |
| Microsoft SQL Server Management Tool | Design database | 2012 |
| GitHub for Windows | Documents and Source Code’s Version Management | 1.0.48.0 |
| Gliffy Online | An account on Gliffy .com for designing screens |  |
| Adole Photoshop CS5 | Design icon, button, logo. |  |

**Table B-I-5-1: Tools**

1. **Techniques**

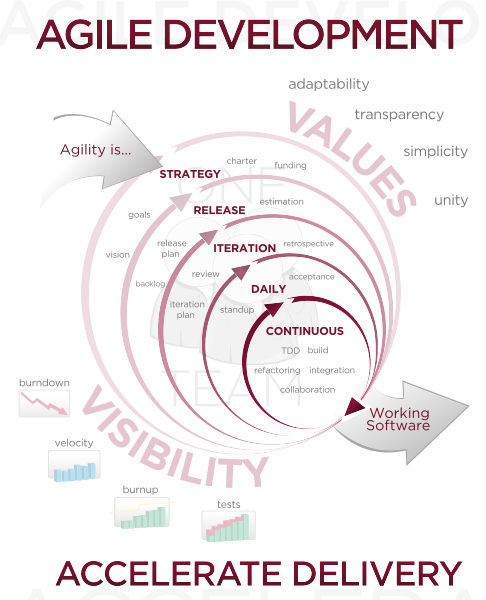
|  |  |
| --- | --- |
| **Technical** | **Description** |
| WCF Services | Visual Studio is the greatest IDE for developing WCF Services. Because our team has experience and skill in C# so we strongly recommend that C# is programming language. |
| Database | We chose Microsoft SQL Server for database development. This database will be deployed to Microsoft Azure. |
| Android application | Java is our favorite programming language and Eclipse is freedom IDE. We decide use Android Platform for Fat Client development |

**Table B-I-5-2: Techniques**

1. **Project Organization**

### Software Process Model

Because this system is developed from scratch, and requires many changes and updates, we have chosen the Agile Software Development for project’s Software Process Model



**Benefits of Agile Software Development**

* **Mitigate Risk**: The problem with risk is that sometimes you see it coming…and sometimes you don’t. In many cases, perceived risks to a project may prove not to be risks at all while unanticipated risks may surface out of the blue. Because many risks are only addressed and discovered during integration, an iterative approach provides a far better opportunity to mitigate them early on because the iterative approach is a near-continuous integration.  What used to be long, uncertain, and difficult to plan accurately (taking up to 40% of the total effort at the end of a project) is divided into 6-9 smaller integrations that start with far fewer elements.
* **Accommodate Change**: Changes in requirements and scope have always been primary sources of trouble for a project, leading to late delivery, missed schedules, and unsatisfied customers. But the iterative approach takes changing requirements into account from the start. We have to expect users to change their minds as the project evolves—it’s inevitable. And they’re right to do it. After all, the context is changing. As they learn more about the environment, technology, and their own business, and see intermediate demonstrations of the product as it is being developed, this additional knowledge contributes to new ideas/requirements and thereby fosters a more comprehensive business solution in the end. Forcing users to accept the system as originally imagined is not only unrealistic, but detrimental to true success. Iterations also allow for technological change. If technology changes or new technology appears, the project can take advantage of it. This is particularly relevant for platform changes and lower-level infrastructure changes
* **Achieve Higher Quality:** An iterative approach results in a more robust architecture. Flaws are detected and corrected earlier in the project lifecycle. Performance bottlenecks discovered early can be reduced, as opposed to being discovered on the eve of delivery. Testing iteratively, as opposed to executing acceptance testing toward the end of the project, results in a more thoroughly tested product. And because critical functions have had many opportunities to be tested over several iterations, the result is more mature testing methodologies and higher quality software.
* **Learn, Improve, Foster Buy-In:** An iterative approach provides opportunities for the entire project team to enhance their knowledge and skills throughout the project lifecycle, contributing to the ultimate success of the project. It provides more opportunities for lessons learned. Testing starts early, technical writing starts early, and developers start coding sooner. The result is a better system supported by better training and help materials, making for happier and more accepting end users. Lastly, the need for additional training or resources can be detected in early iteration assessment reviews. After all, why put off till tomorrow what is better accomplished today?

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Full Name** | **Team Role** | **Responsibilities** |
| 1 | Vũ Duy Khánh | Project Manager, Developer | Process Managing  Database Designing  Collecting and Developing Documents |
| 2 | Lương Anh Sơn | Member, Designer | Software Designing  Documents |
| 3 | Nguyễn Trung Dũng | Member, Developer | Developing Client side  Documents |
| 4 | Lương Thanh Hùng | Member, Developer | Developing Server side  Documents |
| 5 | Phạm Minh Hoàng | Member, Tester | Testing  Documents |

**Table B-II-2-1: Roles and Responsibilities**

1. **Project Management Plan**

### Tasks

1. **Initiating**

|  |  |
| --- | --- |
| Description | Identify project, project team, stakeholder, project’s purpose and technical solution |
| Output | Introduction document |
| Deliverables | Report No.1 |
| Dependencies and Constraints | N/A |
| Risks | Misunderstand in project identifying, lack of knowledge about new technical |

**Table B-III-1-1: Initiating**

1. **Planning**

|  |  |
| --- | --- |
| Description | Plan to manage human resources, define the project goals and objectives, identify tasks and how goals will be achieved, quantify the resources needed, and determine budgets and timelines for completion. |
| Output | Planning document, schedule |
| Deliverables | Report No.2, MPP file |
| Dependencies and Constraints | N/A |
| Risks | Under schedule or over schedule |

**Table B-III-1-2: Planning**

1. **Technical Training**

|  |  |
| --- | --- |
| Description | Study the basics of developing in Android, Cloud Computing and WCF Services |
| Output | Samples about Android, WCF Services |
| Deliverables | N/A |
| Dependencies and Constraints | N/A |
| Risks | The technologies may be new and difficult to learn |

**Table B-III-1-3: Technical Training**

1. **Software Requirements Analysis**

|  |  |
| --- | --- |
| Description | Analyze software requirements to create software requirements specification document |
| Output | Software Requirement Specification document |
| Deliverables | Report No.3 |
| Dependencies and Constraints | N/A |
| Risks | N/A |

1. **Database Design**

|  |  |
| --- | --- |
| Description | Identify main tables, columns and type of value. Draw entity relationship diagram |
| Output | Database detail |
| Deliverables | N/A |
| Dependencies and Constraints | N/A |
| Risks | Under schedule or over schedule |

1. **User Interface Design**

|  |  |
| --- | --- |
| Description | Design user interface base on use cases, draw work-flow screen, define actions on each screen |
| Output | User Interface Design |
| Deliverables | N/A |
| Dependencies and Constraints | N/A |
| Risks | Under schedule or over schedule  User Interface does not meet customer requirements |

1. **Software Detail Design**

|  |  |
| --- | --- |
| Description | Chose System Architecture, draw components diagram, class diagram, sequence diagram |
| Output | System Detail Descriptions |
| Deliverables | Report No.4 |
| Dependencies and Constraints | N/A |
| Risks | Under schedule or over schedule |

1. **Develop Server Side Application**

|  |  |
| --- | --- |
| Description | Develop WCF Services (Task Management Services, User Management Services, Message and Notification Services) |
| Output | Source code |
| Deliverables | Executable program and source code |
| Dependencies and Constraints | N/A |
| Risks | Lack of knowledge about WCF Services Developments  Under schedule or over schedule |

1. **Develop Client Side Application**

|  |  |
| --- | --- |
| Description | Develop Android Application (Task Management Functions, User Management Functions, Message and Notification Functions) |
| Output | Source code |
| Deliverables | Executable program and source code |
| Dependencies and Constraints | N/A |
| Risks | Lack of knowledge about Android Application  Under schedule or over schedule |

1. **Testing**

|  |  |
| --- | --- |
| Description | Creating test case and execute test |
| Output | Test plan, test case document, test report, all tested modules and tested system |
| Deliverables | Test documentations (Test plans, test reports) |
| Dependencies and Constraints | Developments are done |
| Risks | Under schedule or over schedule  Unit test may not be performed thoroughly causing spending many efforts in system test phase. |

### Task Sheet: Assignment and Timetable

TBD (To be determined)

1. **Coding Convention**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description** | **Detail, Reason & Example** | **How to use** |
| 01 | Use the default Code Editor settings in Visual Studio 2012 | **Reason:** Easy to setup and control source code  **Example**: Smart indenting, four-character indents, tabs saved as spaces… | Tools, Options, Text Editor, C# Formatting |
| 02 | Use the default Code Editor settings in Eclipse | **Reason:** Easy to setup and control source code  **Example**: appearance color, annotations… | Windows, Preferences, General, Text Editors |
| 03 | Display line number | **Reason:** It helps on work effort management and debugging process  **Example**: appearance color, annotations, | Check on display line number on Code Editor Settings |
| 04 | Don’t Ignore Exception | **Reason:** It helps on debugging and error controlling  **Example :**  **try**{  //Doing something  }**catch**(IOException i){  //Error Message  }**Finally**{  } | N/A |
| 05 | Don’t catch generic Exception | **Reason:** Programmers have to know exactly what type of Exception and where to catch Exception  **Example:**  **try**{  someComplicateIOFunction();  // may throw or catch IOException  loadData();  // may throw or catch SqlException  }**catch**(IOException i){  //Error Message    }**finally**{  } | N/A |
| 06 | Full qualify imports | **Reason:** save the time to fix import error, reduce the number of import statements and improve effort  **Example:**  **import** android.widget.\*; | N/A |
| 07 | Create a variable for iterating before the loop | **Reason:** improve program performance  **Example:**  Wrong  **for** (x = 0; x < rows.count(); x++) {  // Doing something  }    Right  **int** rowsNum = rows.count();  **for** (x = 0; x < rowsNum; x++) {  // Doing something  }  In the wrong example, every loop the count() function will run, which can be very expensive for some collection. In the right example, the count() function will only run once. | N/A |
| 08 | Don’t initialize variable inside loops | **Reason**: improve program performance  **Example:**  Wrong  **for** (x = 0; x < rows.count(); x++) {  User user = **new** User(); System.out.print(user.UserId);  }    Right  User user = **new** User();  **for** (x = 0; x < rowsNum; x++) {  System.out.print(user.UserId);  }  In the wrong example, every loop the User class is initialized, which waste processor and memory. | N/A |
| 09 | Put open brace with preceding | **Reason:** make the code more clearly  **Example:**  **if**(s == "abc" && s == "cba"){  // Doing something  } | N/A |
| 10 | Variable naming convention | **Details:**   * Non-public, non-static field names with m * Static field names start with s * Other fields start with a lower case letter * The name of array variable must be plural noun   **Reason:** Make the code easier to understand  **Example:**  **public** String firstName;  **int** mAge;  **private** **static** **final** String *sMessage*;  **public** List<User> users; | N/A |
| 11 | Method naming convention | **Detail:**   * Name of methods have to start with verb. * Verb begin in upper case if it’s public and lower case or preceded by underscore (\_)   **Reason:** Make the code easier to understand  **Example:**  **public** **void** DisplayMessage(String s) {  }  **private** **void** \_displayMessage(String s) {  } | N/A |
| 12 | Class naming convention | **Detail:** Beginning with upper case  **Reason:** Make the code easier to understand  **Example:** User, Task | N/A |
| 13 | Interface naming convention | **Detail:** Beginning with I character  **Reason:** Make the code easier to understand  **Example:**  IUser | N/A |
| 14 | Use #region and #endregion to tidy up | Reason: Make the code easier to view  Example:  #region  public int Add(int x, int y){  throw new NotImplementedException();  }  #endregion | Visual Studio |
| 15 | Don’t use magic number or raw string, create a constant for them | Reason: Make the code easier to understand  Example:  Wrong  **if** (a == 5) {  b = "This is type a";  }  Right  **const** TYPE\_A = 5;  **const** TYPE\_A\_NOTICE = "This is type a";  **if** (a == TYPE\_A){  b = TYPE\_A\_NOTICE;  } | N/A |
| 16 | Use standard comments | Reason: Helps programmers and reviewer to understand source code  Example:  /\*\* Return the correctly rounded positive square root of a double value.  \*/    **static** **double** sqrt(**double** a) {  } | N/A |

1. **SOFTWARE REQUIREMENT SPECIFICATION**
2. **Introduction**
3. **Objectives**

This is the Software Requirements Specification for the Project manager system. This SRS details the capabilities and functions that the Project manager must be capable of performing Mobile Project Manager is designed to support people to manage their daily tasks, people can share with each task. These requirements will assure that the system will correctly and reliably perform its intended functionality. This specification will provide general, as well as specific requirements to be used in the design, testing and validation of the system.

1. **Scope**

Scope of Software Requirement Specifications Document is defined General User Requirement Specification, External Interface Requirement, function and non-function. This Software Requirements Specification applies to the Project Manager system, including the Project manager Android applications. Mobile Project Manager helps people create and manage tasks for themselves. It also can create team and project team. MPM allow manager can add members to the project team and assign tasks to them

1. **Requirements**
2. **User Requirement Specification**
   1. **Consumer user requirements**

Consumer user details include:

* Personal Detail.
* Contact Detail.
* His/her Task Detail.
* Login Detail.
* Gender
* Type of User

Consumer User is able to view his/ her detail and other user detail.

Personal Detail: Including Display Name and Avatar

* Consumer user can change display name. In the first time his/ her login, display name of user is null and his/her is able to set it in user profile .After that, his/ her is able to change it by the same way. The display name is only imported from Latin or ASCII Characters.
* Consumer user is able to change and set avatar. In the first time his/ her login, or if consumer user do not want to set his/ her avatar, the avatar is default, his/ her is able to set and change it in user profile. The avatar could be imported from image in his/her cell phone or capture by camera. It is depending on consumer user intention.

Contact Detail: Email. That is email which user has registered when sign up.

Login Detail: Allow user to change their password.

Gender: Choosing when user register, male and female user has difficult default avatar.

Type of User: Default type of user is consumer user, he /she is able to change his/ her type user to business admin by sending a request to server and pay cash for it.

Contact: A list of user .User can add some other users to this list to easy to send message with them. User enters a name in text box and some simulation display name is displayed. User chose one to add to this list.

Message and Notification:

* View Message :
* User can see the Notification icon on the home screen. It must be easy to find out it with an envelope symbol. The special here is that display a number of Notifications in the right corner envelope icon with red color. This color is highlighting that user sees first look. The number here is total of new Notifications that user didn’t read before. When user touches to the Notification icon, a drop list will appear with other contents like: “You have a message from Alex”; “Laura has removed from the ABC task”, “Long added you to Project 1’.
* With one kind of Notification has one small separate symbol in the first sentence of Notification. When user read a Notification then the Notification is deleted, user cannot see it again. And the number will be reduced in the envelope icon. Example: The number of Notifications is 100. When you touched a Notification: “You have a message from Alex” as you read. User chooses other function and user sees the number of Notifications is reduced from 100 to 99. And user touches in envelope icon, this notification is disappeared in the drop list. The Notification is deleted”.
* When user touches in message Notification, a message screen display contains 4 parts: name of sender, received message content, reply message content and send button. User types contents to answer the sender and touch reply button. The message is sent.
* Send Message :
* Business admin chooses Project Member function. A member list is displayed. He/she touch in user avatar and drop list will appear with send function and delete function. Choose send function and a send screen open. The send screen contains one function is that user send the message to the user that chose to send before.
* Consumer user is able to send message in Contact screen, he/she touch in user avatar to chose send message function like business admin
* Business admin is able to send message from Contact too.

Task Detail: Including all task which consumer user is assigned, he/she is able to jump to edit task from task detail.

Login:

* Have 2 text fields: Email & Password and 2 button Login & Register
* User must be entering the email & password that registered before.
* User touches login button to access the home screen.
* User touches register button to access Register screen.

Logout: After user login to access the application. User can touch to user symbol in the top of the right corner on home screen. Touch this and drop list display, and choose “logout” and user account get out of application control access.

Registration: Have 3 textbox fields and 1 drop list and 1 button: Email, Password, Confirm Password, Gender and Submit button.

Email textbox field: User enters a standard email.

Password textbox field:

Passwords must contain Latin and ASCII characters

Base 10 digits (0 through 9). No alphanumeric characters: ~! @#$%^&\*\_-+=`|\(){}[]:;"'<>,.?/. Password length is more than 6 characters. Uppercase or lowercase.

Confirm password field: the password is the same with the first pass in password textbox field.

Gender drop list: Gender must choose one of two: Male or female

If user inputs wrong that rules, application request user enter information again until to true rules.

When user enter full information like Email, Pass, Gender touch Submit button. Application will send a confirm message to the user email that entered before.

Project: Consumer user is able to create new projects management and able to delete them. Consumer user could be added to a project by business admin user.

State: Default state is To Do, Doing, Done. Consumer user is able to add new states including name of state and able to edit name of state in State Board Setting screen.

Task:

* Add new task : Consumer user is able to add new task including :
* Name of task: Only imported from Latin and ASCII character. Name of task could be blank.
* Start Date: User set a start date by choosing a date in calendar. Start Date could be blank.
* Due Date: User set a start date by choosing a date in calendar. Due Date could not be blank
* Description: Default description is blank; a task has or has not description depending on user intention.
* Consumer user is able to view task of him/her or other member added to project in task screen, and consumer user is able to edit/update his/her task including:
* Name of Task
* Start Date
* Due Date: Displaying of Due Date would be change. In normal case (remaining along of time until due date) due date is white color, if remaining 24 hours until sue date, it change color into yellow, if due date, it is red color.
* Description
* Move Task: user is able to move task from state to other state by drag/drop a task to other state place.
* Delete Task: consumer user can delete tasks which created by himself/herself. When user deletes a task, a warning message box will be display. After deleting a task, all information of that task will be deletes.
* Comment: After creating a task, consumer user is able to comment in a task by touching comment icon in tasks screen. After touching comment icon, a list of all comment of all users has displayed, consumer user can view all of them, write comment in text box and submit it. After submitting, all other user can see it.
  1. **Business admin requirements**

Consumer user details include:

* Personal Detail.
* Contact Detail.
* His/her Task Detail.
* Login Detail.
* Gender
* Type of User

Type of business user could not be change to consumer user.

Business admin is able to view his/ her detail and other user detail like consumer user.

Project: Business user is able to add other user to his/her project management and manage them in his/her project.

* Adding a member: Business admin can add a member to project in project member management. After chose “Add member” and enter some key words, some user with simulation display name will be displayed, business admin is able to chose one to add to project.
* Deleting a member: After adding member, with some reason (incorrect when adding, member is out of project) business admin is able to remove them in project management.
* Business admin touch in avatar of the user he/she want to remove, and chose remove from project. If user added to project and assigned tasks, when business admin remove them from project, all assigned tasks of them in that project will be disappear, and in task screen, assigned user list will not display them anymore.

State: Default state is To Do, Doing, Done. Business admin is able to add new states including name of state and able to edit name of state in State Board Setting screen like consumer user.

Task: Business admin is able to view, edit task, move, delete and comment task like consumer user. In addition he/she is able to assign members to a task.

* Assign member: In adding task screen, assigning member could be blank or not, business admin is able to edit it in edit screen. All of user added to project will be displayed in assigning screen, business admin will check in a member to assign a member to a task.
* Remover member: Business admin is able to remove member from a task. He/she will un-check in a user that checked before to remove a user from a task.

Business admin is able to comment in a task view and send message, notification like consumer user.

1. **System Requirement**
2. **User Interfaces**

* The software interface should be intuitive, friendly easy to use
* Mobile project manager have to support Gestures feature that allow users to interact with your app by manipulating the screen objects you provide such as long press. Swipe, pinch open, pinch close
* The item, logical button layout, easy to understand, not glitzy users, because the use of multiple small mobile screen should be able to press the wrong button

1. **Hardware Interfaces**

* The system should be deployed online and available to a wide range of hardware devices. The system should be accessible on any mobile or tablet using android.
* The system should be Supporting Different Screens; Android categorizes device screens using two general properties: size and density. You should expect that your app will be installed on devices with screens that range in both size and density. As such, you should include some alternative resources that optimize your app’s appearance for different screen sizes and densities

1. **Software Interfaces**

* Programs should run on all machines using android operating system.
* Software should support multiple versions of android.
* Software interface supports English and Vietnamese.

1. **Communication Protocol**

* MPM system uses standard protocols (JSON) and TCP/IP to communicate between clients and server.

1. **Non-Functional Requirement**
2. **Performance**

* Sign-in time should be equal or less than 2 seconds.
* Maximum delay in toggling foreground / background is less than 1 second.
* 1000 concurrent users use the service access at the same time.
* Response time is less than 2 seconds.

1. **Usability**

* Users do not need any tutorial or guide for using this app. They have spent less more than 15 minutes to setup their first project management.
* Support multi languages such as Vietnamese and English.

1. **Scalability**

* Supporting Different Devices: Support all Android devices come in many shapes and sizes all around the world.
* Supporting Different Screens: My app will be installed on devices with screens that range in both size and density. 4 generalized sizes: small, normal, large, xlarge. 4 generalized densities: low, medium, high, extra high.
* Supporting Different Platform Versions: While the latest versions of Android often provide great APIs for my app, I will continue to support older versions of Android until more devices get updated.
* Support Many Themes: Fonts, Colors, Buttons

1. **Maintainability**

* Make future maintenance easier, or cope with a changed environment
* Independence components development and easy for debugging and deploying to meet new customer requirements

1. **Availability**

* The system should be able to available 95% of time.
* The database servers should be available 24 hours/day.

1. **Security**

* Ensure security for user account, not be lost their information
* Authentication is required when accessing into or querying from database system

1. **Functional Requirement**
2. **List of Functions**

**Login Function**

|  |  |
| --- | --- |
| **Code** | **Description** |
| FR01 | User registration |
| FR02 | Login by User ID |
| FR03 | Login by Gmail |
| FR04 | User Logout |

**User Management Function**

|  |  |
| --- | --- |
| **Code** | **Description** |
| FR05 | Add Member to Project/Organization |
| FR05 | View Member |
| FR07 | Remove Member from Project |
| FR08 | View Tasks Member |
| FR09 | Add Contact Member |
| FR10 | Remove Contact Member |
| FR11 | Set and change Display Name |
| FR12 | Set avatar |

**Task Management Function**

|  |  |
| --- | --- |
| FR13 | Create Project |
| FR14 | View Project |
| FR15 | Delete Project |
| FR16 | Create Task |
| FR17 | Assign Task Member |
| FR18 | Edit Task Member |
| FR19 | Comment in Task |
| FR20 | Change Task Status |
| FR21 | Move Task |

**Message Function**

|  |  |
| --- | --- |
| FR22 | Send Message to a member |
| FR23 | Send Message when a task over due |
| FR24 | Send Message when making a event |

1. **Function Description**

FR01: Login by Gmail

Access right: Users

Description: Users use this function to access application by Gmail

FR02: Login by User ID

Access right: Users

Description: Users use this function to access in application by use a id has registered before

FR03: User Registration

Access right: Users

Description: Users use this function to register, an user become a member to use application

FR04: Add member

Access right:

Description: use this function to add a username to database of project management.

FR05: View Member

Access right: Users

Description: Users use this function to view all public information of a user

FR06: View Task of Member

Access right: Users

Description: Users use this function to view all assigned tasks of a member

FR07: Remove user from project

Access right:

Description: use this function to remove a exist member from projects.

FR08: Create a group member

Access right:

Description: use this function to organize members by making a group

FR09: Add member to group

Access right:

Description: use this function to manage their project.

FR10: Remove member from group

Access right:

Description: use this function to remove a exist member from group

FR11: Add group to Project

Access right:

Description: use this function to add a group of member to Project

FR12: Delete a group

Access right:

Description: use this function to delete a exist group.

FR13: Send message to a member

Access right: Users

Description: Users use this function to communicate between 2 of members

FR14: Send message to a group member

Access right: Users

Description: Users use this function to send a message to all members of group

FR15: Send message when a task over due

Access right: Users

Description: Users use this function to remind users about overdue task.

FR16: Send message when a task at the approach of over due

Access right: Users

Description: Users use this function to manage their sessions by logging in or out of the system, which grant them the rest of the system’s functions.

FR17: Send message when add a member

Access right: Users

Description: Users use this function to remind users about adding to a project

FR18: Send message when remove a member

Access right:

Description: use this function to remind users about removing from a project

FR19: Send message when assign a task to a member

Access right:

Description: use this function to remind members about assigned a task in project.

FR20: Send message when removing assign a task to a member

Access right:

Description: use this function to remind members about removed assign a task in project

FR21: Send message when change complete a task

Access right:

Description: use this function to remind members when a member complete a task

1. **User Case**
2. **Main use case**

Chia ra thanh cac Group User Cases, e.g: Task Management, Account Management…

****

*Functional specifications*

1. **Task Management Function**



|  |  |
| --- | --- |
| **Use Case ID** | **T-001** |
| **Use Case Name** | **1-Create Project** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | The application will be make a name to the list for uses to manage their project |
| Actors | Users |
| System | Mobile Project Management |
| Preconditions | User has already logged in the system |
| Post-conditions | The application will be displayed name of project on project’screen. |
| Flow of Events: Create Project | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Create project” button | Dialog box “Create project ” appear |  |  | | 2 | Enter the project name |  |  |  | |  | Touch “Save” button | Display project name on project screen | |
| Exception | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Not enter the project name |  |  |  | | 2 | Touch “Save” button | “Save” button is invisible |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-002** |
| **Use Case Name** | **2-View Project** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | This function allows users view an existing project that was created before. |
| Actors | Users |
| Preconditions | Access in application and touch on a existing project |
| Post-conditions | Display state board and tasks on state board screen |
| Flow of Events: View Project | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch one of the list project | Display state board and tasks in project |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-003** |
| **Use Case Name** | **3-Create Task** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | The application will create task in project |
| Actors | Users |
| System | Mobile Project Management |
| Preconditions | Project has already created |
| Post-conditions | The application will be displayed task on state board |
| Flow of Events : Create Task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Add Task” button | “Add Task ” screen appear |  |  | | 2 | Enter the task name |  |  |  | | 3 | Enter the description |  | | 4 | Choose start date | DatePicker UI appear | | 5 | Choose due date | DatePicker UI appear | | 6 | Touch “Create” button | Task is created and displayed on stateboard | |
| Exception | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Not enter the task name | Warning text display |  |  | | 2 |  | “Create” button is invisible |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-004** |
| **Use Case Name** | **4 – Move task** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | This function allows users move tasks between state boards |
| Actors | Users |
| Preconditions | Task has created before |
| Post-conditions | Task is moved to a new state board |
| Flow of Events: Move task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch and hold a task at least 1 second | Task has been selected light up |  |  | | 2 | Drag and drop task to a new state board | Task has been moved to a new state boad | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-005** |
| **Use Case Name** | **5 – Edit task** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | This function allows users edit an existing task on project |
| Actors | Users |
| Preconditions | Task has created before |
| Post-conditions | Task has updated new infomation |
| Flow of Events: Edit task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a task on stateboard | “Edit Task” screen appear |  |  | | 2 | Change the task information |  | | 3 | Touch button Save | The new information of task is update | |
| Exception | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Step | Actor Input | System Response | |  |  | | 1 | Make the task name blank | | Warning text display |  |  | | 2 |  | | “Save” button is invisible |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-006** |
| **Use Case Name** | **6 – Assign Task** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | This function allows users assign a task to member of project |
| Actors | Users |
| Preconditions | Task has created before |
| Post-conditions | Task is assign to member |
| Flow of Events: Assign Task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a task on stateboard | “Edit Task” screen appear |  |  | | 2 | Touch button “Assign to” | “Assign task” screen appear | | 3 | Choose members from member list to assign |  | | 4 | Touch Assign button | “Assign task” screen close | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-007** |
| **Use Case Name** | **7 – Comment in task** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | This function allows users comment in a task |
| Actors | Users |
| Preconditions | Task has created before |
| Post-conditions | Comment will be display on comment board |
| Flow of Events:Comment in task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a comment icon on task card | “Comment task” dialog box display |  |  | | 2 | Comment on textbox |  | | 3 | Touch button “Post” | The comment is post to comment board on task card | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-008** |
| **Use Case Name** | **8 – View my task** |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | This function allows users view task you have assigned |
| Actors | Users |
| Preconditions | User have assigned at least 1 task |
| Post-conditions | Sort tasks’ userhas assigned |
| Flow of Events:View my task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch member manage icon on project screen | Member mage screen is appear |  |  | | 2 | Touch button “View my task” | The list of task’ user has assigned display | |

1. **User Management Function**



|  |  |
| --- | --- |
| **Use Case ID** | **U-001** |
| **Use Case Name** | **Login by Gmail** |
| Date | 5/31/2013 |
| Author | HungLT |
| Brief Description | This function allows to members access application by Gmail |
| Actors |  |
| System | Google service |
| Preconditions | Access in application |
| Post-conditions | Display projects management screen. |
| Flow of Events : Login by Gmail. | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Login by Gmail” button. | Display a sub box. |  |  | | 2 | Use Gmail to login | Google service response |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-002** |
| **Use Case Name** | **Login by User ID** |
| Date | 5/31/2013 |
| Author | HungLT |
| Brief Description | This function allows to members access in application by use a id has registered before. |
| Actors |  |
| System | Mobile Project Management |
| Preconditions | Access in application. User logged out or has not login to system yet . |
| Post-conditions | Display projects management screen. |
| Flow of Events : Login  By User ID | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Enter ID, l and Password correctly. | Access in application successful |  |  | |
| Exceptions: | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Do not enter ID or Password correctly. | Login fail, display warning message |  |  | | Do not enter ID or Password | Login fail, display warning message |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-003** |
| **Use Case Name** | **User Logout** |
| Date | 6/3/2013 |
| Author | HungLT |
| Brief Description | This function allows to members access application by Gmail |
| Actors |  |
| System | Mobile Project Management. |
| Preconditions | Access in application |
| Post-conditions | Display projects management screen. |
| Flow of Events : User Logout | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Access in project management screen |  |  |  | | 2 | Touch user icon seriatim in 2 times | Display a sub drop down box. |  |  | | 3 | Touch Logout | User access in application no more. |  |  |  |  |  |  | | --- | --- | --- | | Step | Actor input | System response | | 1 | Access in “Organize” screen |  | | 2 | Touch “ Log out” | User access in application no more. | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-004** |
| **Use Case Name** | User registration |
| Date | 5/31/2013 |
| Author | HungLT |
| Brief Description | This function allows registering a user to use application. |
| Actors |  |
| System | Mobile Project Management |
| Preconditions | Access to Sign up screen or access in to Login screen, then touch “Create an Account” button . |
| Post-conditions | A user was created successful, a mail has been sent to email of registration user. |
| Flow of Events : User registration | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Enter ID, Email and Password correctly. |  |  |  | | 1 | Touch “Create” | A mail has been sent to email of registration user. |  |  | |
| Exceptions: | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Create an Account” button. | Display a sub box. |  |  | | 2 | Do not enter name, or password, or email | Display warning message, a confirmation  Email cannot be send. |  |  | | Do not enter or wrong input in password confirmation | Display warning message, a confirmation  Email cannot be send. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-005** |
| **Use Case Name** | **Add Member** |
| Date | 5/29/2013 |
| Author | HungLT |
| Brief Description | The main function for a group to manage their project. |
| Actors |  |
| System | Mobile Project Management |
| Preconditions | Touch User icon in project management screen. |
| Post-conditions | The application will be add a user name to database of project management. |
| Flow of Events : Add User | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Step | Actor Input | System Response | |  |  | | 2 | Select “Add Member” | Display a text box. | | 3 | Enter a user name | Display some similar user name in drop list | | 4 | Touch a user name in list | A user name will be added to database | |
| Exceptions : | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Step | Actor Input | System Response | |  |  | | 1 | Select “organize member” | Display project manage ‘screen | |  |  | | 2 | Select “Add Member” | Display a text box. | | 3 | Do not enter a user name | Drop list does not appear | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **U-006** | |
| **Use Case Name** | **View Member** | |
| Date | 5/29/2013 | |
| Author | HungLT | |
| Brief Description | This function allows users view all public information of a user. | |
| Actors | Users | |
| System | Mobile Project Management |
| Preconditions | User is added to Project or assigned tasks. | |
| Post-conditions | The application will be displayed information of user | |
| Flow of Events : View | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a user icon | Display information of user in sub box |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **U-007** | |
| **Use Case Name** | **View Task Member** | |
| Date | 5/29/2013 | |
| Author | HungLT | |
| Brief Description | This function allows users view all assigned tasks of a member | |
| Actors | Users | |
| System | Mobile Project Management |
| Preconditions | User is added to Project or assigned tasks. | |
| Post-conditions | The application will be displayed information of user | |
| Flow of Events : View Task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a user icon | Display information of user in sub box |  |  | | 2 | Touch “<” button on sub box | Display assigned tasks of member |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **U-008** | |
| **Use Case Name** | **Remove Member from Project** | |
| Date | 5/30/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remove a member that was added from project. | |
| Actors |  | |
| Preconditions | Access in project member management ‘screen  User was added to Project. | |
| Post-conditions | User removed from Project, if user was assigned one or more tasks before, member will be removed from list member of that tasks | |
| Flow of Events : Remove Member from Project | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a user icon | Display information of user in sub box |  |  | | 2 | Touch ”Remove from Project” | Display message “are you sure”. |  |  | | 3 | Touch “Yes” | Member was removed from Project. |  |  | | |
| Exceptions : | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a user icon | Display information of user in sub box |  |  | | 2 | Touch ”Remove from Project” | Display message “are you sure”. |  |  | | 3 | Touch “No” | Member was not removed from Project. |  |  | | |

1. **Message Management Function**



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-001** | |
| **Use Case Name** | **Send Message to Member** | |
| Date | 5/30/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to communicate between two of members | |
| Actors | Users | |
| Preconditions | User is added to Project or assigned tasks. | |
| Post-conditions | A notification will be sent to a member | |
| Flow of Events : Send message to a member | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a user icon | Display information of user in sub box |  |  | | 2 | Click “Send Message” | Display a sub box |  |  | | 3 | Enter some notes in the message field |  |  |  | | 4 | Touch “Send” | A notification include message will be sent to a member |  |  | | |
| Exception:: | |  |  |  | | --- | --- | --- | | Step | Actor Input | System Response | | 1 | Touch a user icon | Display information of user in sub box | | 2 | Click “Send Message” | Display a sub box | | 3 | Do not enter note in the message field | “Send” button is invisible | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-002** | |
| **Use Case Name** | **Send Message to Group Member** | |
| Date | 5/30/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to send a message to all members were added to group. | |
| Actors | Users | |
| Preconditions | User is added to Project or assigned tasks. | |
| Post-conditions | A notification will be sent to all members | |
| Flow of Events : Send message to a member | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a group on group member list | Display a sub box |  |  | | 2 | Click “Send Message to All” | Display a sub box |  |  | | 3 | Enter some notes in the message field |  |  |  | | 4 | Touch “Send” | A notification include message will be sent to a member |  |  | | |
| Exception:: | |  |  |  | | --- | --- | --- | | Step | Actor Input | System Response | | 1 | Touch a group on group member list | Display a sub box. | | 2 | Click “Send Message to All” | Display a sub box. | | 3 | Do not enter note in the message field | “Send” button is invisible | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-003** | |
| **Use Case Name** | **Send Over Due Task Message** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remind users about over due task. | |
| Actors | Users | |
| Preconditions | A task was set due date | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : Send message **when a task over due** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Do not touch compete on all check list tasks in task at over due time | A notification will be sent to all assigned member and leader. |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-004** | |
| **Use Case Name** | **Send a Message about Warning Over Due Task Message** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remind users about over due task. | |
| Actors | Users | |
| Preconditions | A task was set due date | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : Send message **when a task at the approach of over due** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Do not touch compete on all check list tasks in task when current time is 12 hours left to overdue time | A notification will be sent to all assigned member and leader |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-005** | |
| **Use Case Name** | **Send a Message about Adding Member** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remind users about adding to a project | |
| Actors |  | |
| Preconditions | User has not added yet to Project | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : **Send message when add a member** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Add member/ add group member from Project member management screen | A notification will be sent to all member who are added to project. |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-006** | |
| **Use Case Name** | **Send a Message about Removing a Member** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remind users about removing from a project | |
| Actors |  | |
| Preconditions | User was added to Project | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : **Send message when remove a member** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Remove member from Project member management screen | A notification will be sent to member who is removing to project. |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-007** | |
| **Use Case Name** | **Send a Message about Assign Task** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remind members about assigned a task in project | |
| Actors |  | |
| Preconditions | Member was added to Project  Member has not assigned yet in that task. | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : **Send message when assign a task to a member** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Assigned a task for one or more member in “Assign Task” in Edit Task Screen | A notification will be sent to member who is added to project. |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-008** | |
| **Use Case Name** | **Send a Message about Removing Assign Task** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remind members about removed assign a task in project | |
| Actors |  | |
| Preconditions | Member was assigned a task | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : **Send message when removing assign a task to a member** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a member icon in assigned member list | Display a sub box |  |  | | 2 | Touch “remove assign” | A notification will be sent to member who is removed assign from project |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-009** | |
| **Use Case Name** | **Send a Message about completing a Task** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to remind members when a member complete a task | |
| Actors |  | |
| Preconditions | Members were assigned a task | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : **Send message when a member change status of task** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a “Compete task” check box in Edit Task | A Message has been sent to user |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-010** | |
| **Use Case Name** | **Send a Message about completing a Task** | |
| Date | 5/31/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to sent a notification when another users send a message to user | |
| Actors |  | |
| Preconditions | Members were added to project. | |
| Post-conditions | A notification will be sent to members | |
| Flow of Events : **Send message when a member change status of task** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a “ User” icon | Display a sub box |  |  | | 2 | Send a message | A message and a notification will be sent to that user |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-011** | |
| **Use Case Name** | **View System Message** | |
| Date | 3/6/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to user view all message has been sent by system. | |
| Actors |  | |
| Preconditions | Access in Project list screen or Project Management screen. | |
| Post-conditions | The message has been displayed. | |
| Flow of Events **View System Message** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Notification” icon | Display all messages had been sent by system before. |  |  | | 2 | Touch a message | Display contents of message. |  |  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **M-012** | |
| **Use Case Name** | **View User Message** | |
| Date | 3/6/2013 | |
| Author | HungLT | |
| System | Mobile Project Management |
| Brief Description | This function allows to user view all message has been sent by another user. | |
| Actors |  | |
| Preconditions | Access in Project list screen or Project Management screen. | |
| Post-conditions | The message has been displayed. | |
| Flow of Events **View User Message** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Message” icon | Display all messages had been sent by users before. |  |  | | 2 | Touch a message | Display contents of message. |  |  | | |

1. **DESIGN & IMPLEMENTATION**
2. **TEST DOCUMENTATION**
3. **USER’S MANUAL**
4. **APPENDIX**