

**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

Mobile Project Management

|  |  |
| --- | --- |
| **FS Group** | |
| **Group Members** | Vũ Duy Khánh – 01 – 00437  Lương Anh Sơn – 02 – 01889  Lương Thanh Hùng – 03 – 00831  Phạm Minh Hoàng – 04 – 01336  Nguyễn Trung Dũng – 05 – 00350 |
| **Supervisor** | Cao Xuân Vinh |
| **Ext Supervisor** |  |
| **Capstone Project code** | MPM |

- Hanoi, 07/2013 -

RECORD OF CHANGES

\*A – Added M – Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Effective Date** | **Author** | **A/M/D** | **Change**  **Description** | **Version** |
| 20/05/2013 | KhanhVD | A | Newly Created | 0.1 |
| 25/05/2013 | KhanhVD | A | Add Report No.1 | 0.1 |
| 28/05/2013 | KhanhVD | M | Modify Report No.1 | 0.1 |
| 05/06/2013 | KhanhVD | A | Add Report No.2 | 0.2 |
| 05/06/2013 | KhanhVD | A | Add Report No.3 | 0.3 |
| 14/06/2013 | KhanhVD | M | Modify Report No.02 | 0.4 |
| 02/07/2013 | KhanhVD | M | Modify Report No.03 | 0.4 |
| 10/07/2013 | KhanhVD | A | Add Report No.04 | 0.5 |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |
|  | KhanhVD |  |  |  |

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

THE TABLE OF CONTENTS

[**MINISTRY OF EDUCATION AND TRAINING** 1](#_Toc362258110)

[**A.** **INTRODUCTION** 9](#_Toc362258111)

[**I.** **Project Management System Introduction & History** 9](#_Toc362258112)

[**II.** **Initial Ideal of the Group** 10](#_Toc362258113)

[**III.** **Existing Products& Systems** 10](#_Toc362258114)

[**1.** **2Do: To Do List | Task List** 10](#_Toc362258115)

[**2.** **GTasks: To Do List & Task List** 11](#_Toc362258116)

[**3.** **Air To-Do** 12](#_Toc362258117)

[**4.** **Comparison with our product** 13](#_Toc362258118)

[**1.** **Solution and Improvement** 14](#_Toc362258119)

[**a.** **Mobile Application Development Overview** 14](#_Toc362258120)

[**b.** **Android Platform** 14](#_Toc362258121)

[**c.** **Cloud Computing** 15](#_Toc362258122)

[**2.** **Our purposes** 15](#_Toc362258123)

[**B.** **PROJECT MANAGEMENT PLAN** 16](#_Toc362258124)

[**I.** **Project Overview** 16](#_Toc362258125)

[**1.** **Project Name** 16](#_Toc362258126)

[**2.** **Project Objectives** 16](#_Toc362258127)

[**3.** **Project Scope** 16](#_Toc362258128)

[**a.** **The scope of the system** 16](#_Toc362258129)

[**b.** **Target users of the system** 16](#_Toc362258130)

[**II.** **Project Environment** 17](#_Toc362258131)

[**1.** **Production Environment** 17](#_Toc362258132)

[**2.** **Development Environment** 17](#_Toc362258133)

[**a.** **Hardware Requirements** 17](#_Toc362258134)

[**b.** **Software Requirements** 17](#_Toc362258135)

[**3.** **Tools** 17](#_Toc362258136)

[**a.** **Document tools** 17](#_Toc362258137)

[**b.** **Development Tools** 19](#_Toc362258138)

[**c.** **Resource Control Tools** 20](#_Toc362258139)

[**d.** **Communication Tools** 20](#_Toc362258140)

[**4.** **Project Organization** 21](#_Toc362258141)

[a. Software Process Model 21](#_Toc362258142)

[b. Roles and Responsibilities 22](#_Toc362258143)

[**5.** **Project Management Plan** 22](#_Toc362258144)

[a. Tasks 22](#_Toc362258145)

[**Initiating** 22](#_Toc362258146)

[**Planning** 22](#_Toc362258147)

[**Technical Training** 23](#_Toc362258148)

[**Software Requirements Analysis** 23](#_Toc362258149)

[**Database Design** 23](#_Toc362258150)

[**User Interface Design** 24](#_Toc362258151)

[**Software Detail Design** 24](#_Toc362258152)

[**Develop Server Side** 24](#_Toc362258153)

[**Develop Client Side** 25](#_Toc362258154)

[**Testing** 25](#_Toc362258155)

[b. Task Sheet: Assignment and Timetable 26](#_Toc362258156)

[c. All Meeting Minutes 27](#_Toc362258157)

[**Meeting 1** 27](#_Toc362258158)

[**Meeting 2** 28](#_Toc362258159)

[**Meeting 3** 29](#_Toc362258160)

[**Meeting 4** 31](#_Toc362258161)

[**Meeting 5** 32](#_Toc362258162)

[**Meeting 6** 33](#_Toc362258163)

[**Meeting 7** 34](#_Toc362258164)

[**6.** **Coding Convention** 34](#_Toc362258165)

[**C.** **SOFTWARE REQUIREMENT SPECIFICATION** 38](#_Toc362258166)

[**I.** **Introduction** 38](#_Toc362258167)

[**1.** **Objectives** 38](#_Toc362258168)

[**2.** **Scope** 38](#_Toc362258169)

[**II.** **Requirements** 38](#_Toc362258170)

[**1.** **User Requirement Specification** 38](#_Toc362258171)

[**a.** **Consumer user requirements** 38](#_Toc362258172)

[**User Management** 38](#_Toc362258173)

[**Message and Notification** 39](#_Toc362258174)

[**Task Management** 40](#_Toc362258175)

[**b.** **Business admin requirements** 41](#_Toc362258176)

[**User Management** 41](#_Toc362258177)

[**Message and Notification** 41](#_Toc362258178)

[**Task Management** 41](#_Toc362258179)

[**2.** **System Requirement Specification** 42](#_Toc362258180)

[a. External Interface Requirements 42](#_Toc362258181)

[**User Interfaces** 42](#_Toc362258182)

[**Hardware Interfaces** 42](#_Toc362258183)

[**Software Interfaces** 43](#_Toc362258184)

[**Communication Protocol** 43](#_Toc362258185)

[b. Non-Functional Requirement 43](#_Toc362258186)

[**Performance** 43](#_Toc362258187)

[**Usability** 43](#_Toc362258188)

[**Scalability** 43](#_Toc362258189)

[**Maintainability** 43](#_Toc362258190)

[**Availability** 43](#_Toc362258191)

[**Security** 44](#_Toc362258192)

[c. Functional Requirement 44](#_Toc362258193)

[**Login Function** 44](#_Toc362258194)

[**User Management Function** 44](#_Toc362258195)

[**Project Management Function** 44](#_Toc362258196)

[**Message Function** 45](#_Toc362258197)

[**III.** **User Case** 45](#_Toc362258198)

[**1.** **Main Use Case** 45](#_Toc362258199)

[**2.** **Use Case Specification** 46](#_Toc362258200)

[**a.** **Login Function** 46](#_Toc362258201)

[**b.** **Project Management Function** 51](#_Toc362258202)

[**c.** **User Management Function** 64](#_Toc362258203)

[**d.** **Message Management Function** 72](#_Toc362258204)

[**D.** **DESIGN & IMPLEMENTATION** 79](#_Toc362258205)

[**I.** **Design Overview** 79](#_Toc362258206)

[**a.** **Objective** 79](#_Toc362258207)

[**b.** **Scope** 79](#_Toc362258208)

[**II.** **System Architectural Design** 79](#_Toc362258209)

[**III.** **Component Diagram** 84](#_Toc362258210)

[**IV.** **Detail Description of Components** 85](#_Toc362258211)

[**1.** **Class diagram** 85](#_Toc362258212)

[**a. DataAccessLayer** 85](#_Toc362258213)

[**2.** **Class Description** 88](#_Toc362258214)

[**V.** **Sequence Diagram** 115](#_Toc362258215)

[**1.** **Login Screen** 115](#_Toc362258216)

[**2.** **Sign up Screen** 116](#_Toc362258217)

[**3.** **Add New Project Screen** 117](#_Toc362258218)

[**4.** **Project Member Screen** 118](#_Toc362258219)

[**5.** **Add New Member Dialog** 120](#_Toc362258220)

[**6.** **Member Detail Dialog** 121](#_Toc362258221)

[**7.** **Send Message Screen** 122](#_Toc362258222)

[**8.** **Notification Screen** 123](#_Toc362258223)

[**9.** **Add Task** 124](#_Toc362258224)

[**10.** **Edit Task** 125](#_Toc362258225)

[**11.** **Assign Task Screen** 125](#_Toc362258226)

[**12.** **State Board Setting** 126](#_Toc362258227)

[**13.** **Add State Board Screen** 127](#_Toc362258228)

[**VI.** **User Interface Design** 127](#_Toc362258229)

[**VII.** **Database Design** 135](#_Toc362258230)

[**1.** **Entity Relationship Diagram (ERD)** 135](#_Toc362258231)

[**2.** **Table Description** 135](#_Toc362258232)

[**E.** **TEST DOCUMENTATION** 142](#_Toc362258233)

[**I.** **Test Plan** 142](#_Toc362258234)

[1. Features to be tested 142](#_Toc362258235)

[a. Registration function 142](#_Toc362258236)

[b. Log in/ log out function 142](#_Toc362258237)

[c. User management function 142](#_Toc362258238)

[d. Roles function 142](#_Toc362258239)

[e. Task management function 142](#_Toc362258240)

[f. Project management function 142](#_Toc362258241)

[g. State management function 142](#_Toc362258242)

[h. Notification management function 142](#_Toc362258243)

[i. Message management function 142](#_Toc362258244)

[j. Comment function 142](#_Toc362258245)

[k. All Screens 142](#_Toc362258246)

[2. Features not to be tested 142](#_Toc362258247)

[3. Performance 143](#_Toc362258248)

[4. User Acceptance Testing 143](#_Toc362258249)

[**I.** **Test Case** 143](#_Toc362258252)

[1. Registration function 143](#_Toc362258253)

[2. Log in/ log out function 144](#_Toc362258254)

[3. User management function 145](#_Toc362258255)

[4. Project member management function 147](#_Toc362258256)

[5. Project management function 150](#_Toc362258257)

[6. Stateboard management function 151](#_Toc362258258)

[7. Message management function 154](#_Toc362258259)

[8. Notification management function 156](#_Toc362258260)

[9. Task management function 157](#_Toc362258261)

[10. Comment management function 162](#_Toc362258262)

[**II.** **Checklists** 163](#_Toc362258263)

[**F.** **USER’S MANUAL** 163](#_Toc362258264)

[**I.** **Installation Guide** 163](#_Toc362258265)

[**II.** **User’s Guide** 163](#_Toc362258266)

[**G.** **APPENDIX** 163](#_Toc362258267)

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 show 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)** |
| Manage Tasks | X | X | X | **X** |
| Drag & Drop Task | X |  |  | **X** |
| Share task |  |  |  | **X** |
| Comment on task |  |  |  | **X** |
| Send message to other user |  |  |  | **X** |
| Sync with google tasks |  | **X** |  |  |
| User ‘profile |  |  |  | **X** |
| State of task is Changeable |  |  |  | **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**. Cloud Computing Provider such as Amazon, Microsoft… support maintenance tools for user to easy backup and rollback their works

**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**
2. **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.

1. **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 Hardware Requirement | |
| Platform Version | Android OS version 3.0 and higher. |
| CPU | 1 GHz, AML8726-M3 Cortex A9 ARM-based |
| GPU | MALI-400 3D |
| RAM | 512 MB |
| Internal Storage | 8GB Flash |
| Multi –Touch | Yes |
| Camera | Yes |
| Connection | Wireless and 3G |

**Table B-II-1: Minimum Configuration Hardware Requirement**

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.

1. **Tools**
2. **Document tools**

|  |  |
| --- | --- |
| C:\Users\hoangpm01336\Desktop\images (3).jpg | Microsoft Office Project 2007 |
| Project management software, support for planning, assigning task, control progress… |
| Obtained from <http://www.microsoft.com/en-us/download/details.aspx?id=588> |
| **C:\Users\hoangpm01336\Desktop\tải xuống (5).jpg** | Microsoft Word 2010 |
| Use make reports, create charts, draw diagram and make presentations |
| Obtained from <http://www.microsoft.com/vi-vn/download/office-2010.aspx?q=office+2010> |
| C:\Users\hoangpm01336\Desktop\tải xuống (6).jpg | Microsoft Excel 2010 |
| Use make reports, create charts, draw diagram and make presentations |
| Obtained from <http://www.microsoft.com/vi-vn/download/office-2010.aspx?q=office+2010> |
| C:\Users\hoangpm01336\Desktop\images (1).jpg | Microsoft Visio 2010 |
| Use make reports, create charts, draw diagram and make presentations |
| Obtained from <http://www.microsoft.com/vi-vn/download/office-2010.aspx?q=office+2010> |
| C:\Users\hoangpm01336\Desktop\images (2).jpg | Microsoft PowerPoint 2010 |
| Use make reports, create charts, draw diagram and make presentations |
| Obtained from <http://www.microsoft.com/vi-vn/download/office-2010.aspx?q=office+2010> |

**Table B-II-3-1: Document Tools**

1. **Development Tools**

|  |  |
| --- | --- |
|  | Visual Studio Express for Web 2010 |
| WCF Service development purpose |
| Obtained from <http://www.microsoft.com/visualstudio/eng#products/visual-studio-express-for-web> |
|  | Eclipse 4.2.2 |
| Android application development purpose |
| Obtained from <http://www.eclipse.org/downloads/packages/eclipse-classic-422/junosr2> |
|  | Gliffy online |
| A tool for Designing screen |
| Obtained from <http://gliffy.com> |
|  | Adobe Photoshop CS3 |
| A tool for Designing icon, button, logo |
| Obtained from <http://adobe.com> |
|  | Microsoft SQL Server Management Tool 2012 |
| Design database |
| Obtained from <http://www.microsoft.com/en-us/download/details.aspx?id=29062> |

**Table B-II-3-1: Development Tools**

1. **Resource Control Tools**

|  |  |
| --- | --- |
|  | GitHub for Windows 1.0.48.0 |
| Documents and Source Code’s Version Management |
| Obtained from <http://github.com/> |
|  | Dropbox |
| Dropbox is a free service that bring the photos, docs, and videos anywhere and share them easily |
| Obtained from <http://www.dropbox.com> |

**Table B-II-3-3: Resource Control Tools**

1. **Communication Tools**

|  |  |
| --- | --- |
|  | Skype |
| Share the tasks for each other by group chat box. |
| Obtained from <http://www.skype.com> |
|  | FPT Email |
| Contact together with team members. |
| Obtained from <http://mail.fpt.edu.vn> |
|  | Mobile Phone |
| Tool for contact with other members |
|  |
|  | TeamViewer |
| This computer controls the other computer |
| Obtained from <http://www.teamviewer.com> |

**Table B-II-3-4: Communication Tools**

1. **Project Organization**

#### Software Process Model

Because this system is developed from scratch, and requires many changes and updates, we have chosen the Iterative Development Model to base on and develop the system. This approach interleaves the activities of specification, development and validation. The initial system is rapidly developed from very abstract specifications which are provided from the initial customer. Then this is refined with customer input to produce a system that satisfies the customer’s needs.



**Figure B-III-1: Iterative development model**

In term of project management process, we have to perform the following phases:

* **Initial planning:** Use UML tool like VP UML to model and design the abstract system, helping the stakeholders to understand the whole system and each other. Define technical solutions and project purposes.
* **Planning:** Define detail project, scope, project environment, how to organize team working and task assignment and timetable
* **Requirements:** User requirement specification, functional and non-functional requirements and project’s user cases
* **Analysis & Design:** Analyze requirements, design model and functions for the system.
* **Implementation:** Develop the system based on what was designed and analyzed.
* **Testing:** Test the functions of the system, also test the non-functional requirements.
* **Evaluation:** Evaluate all the works that are done to the point, then decide what to do next
* **Deployment:** The final stage: Deploying and packaging the system**.**
* **Termination:** Close out the project.

#### 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

##### **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-IV-1-1: Initiating**

##### **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-IV-1-2: Planning**

##### **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-IV-1-3: Technical Training**

##### **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 |

**Table B-IV-1-4: Software Requirement Analysis**

##### **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 |

**Table B-IV-1-5: Database Design**

##### **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 |

**Table B-IV-1-6: User Interface Design**

##### **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 |

**Table B-IV-1-7: Software Detail Design**

##### **Develop Server Side**

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

**Table B-IV-1-8: Develop Server Side**

##### **Develop Client Side**

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

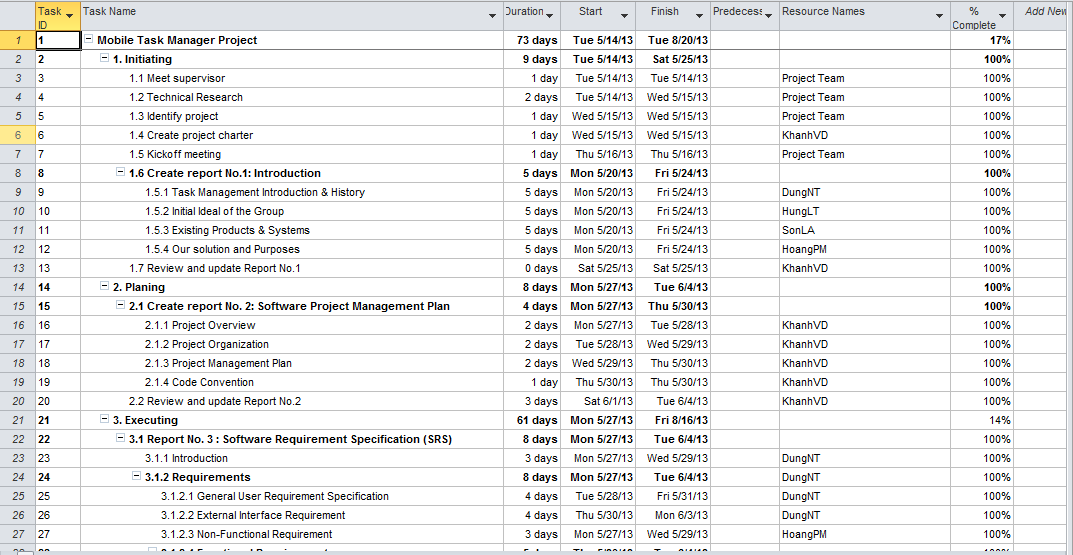
**Table B-IV-1-9: Develop Client Side**

##### **Testing**

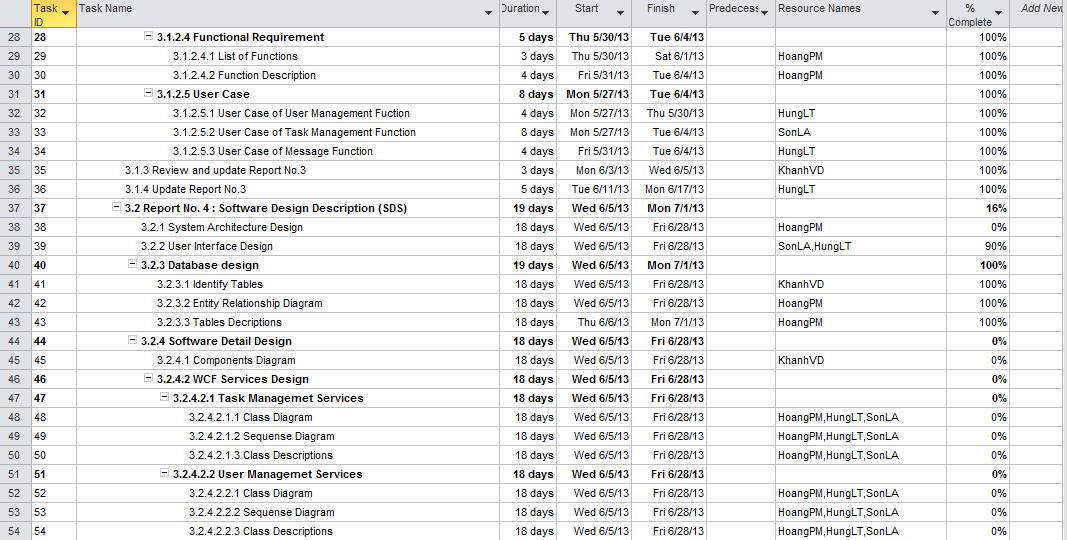
|  |  |
| --- | --- |
| Description | Creating test case and execute test |
| Output | Test plan, test case document, test report, all tested modules and tested system |
| Deliverables | Test documentations (Test plans, test reports) |
| Dependencies and Constraints | Developments are done |
| Risks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

**Table B-IV-1-10: Testing**

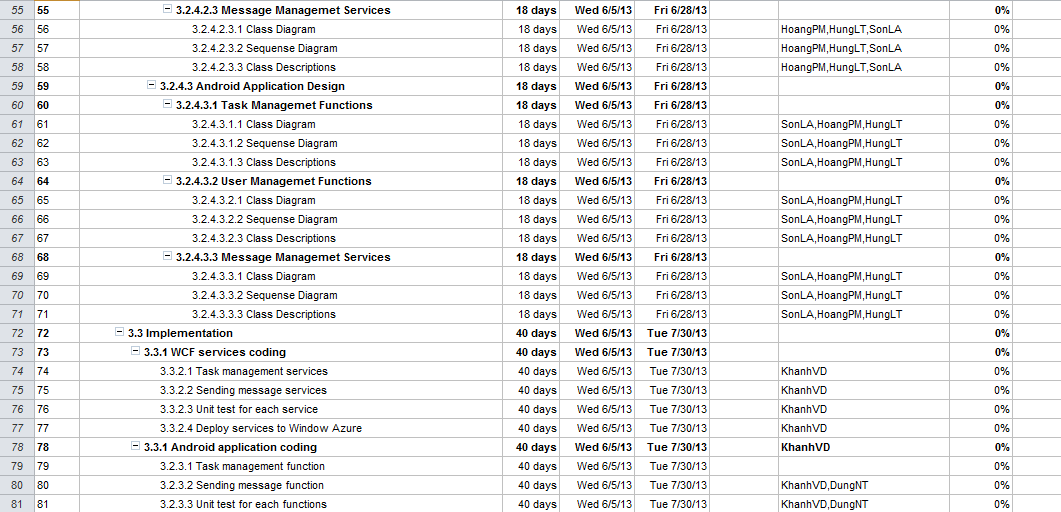
#### Task Sheet: Assignment and Timetable



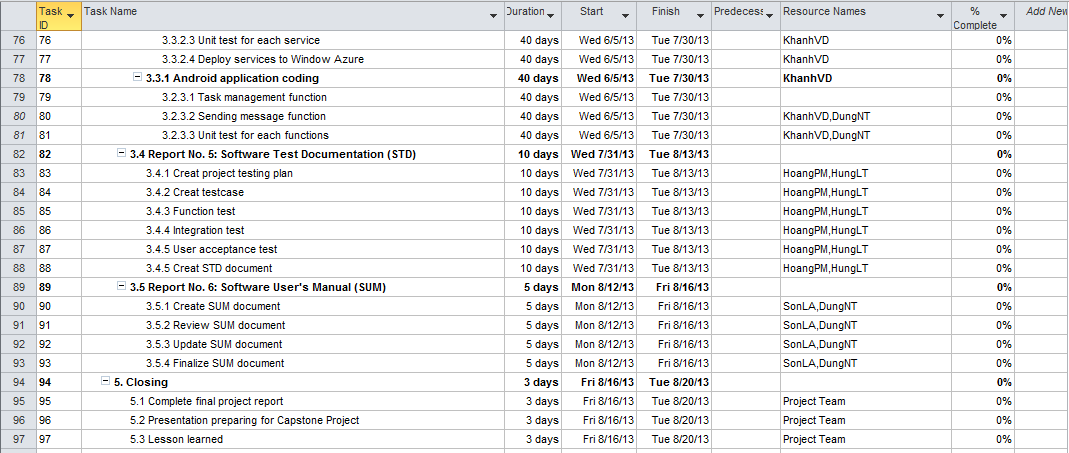
**Table B-IV-2-1: Project Plan 1**



**Table B-IV-2-2: Project Plan 2**



**Table B-IV-2-3: Project Plan 3**



**Table B-IV-2-4: Project Plan 4**

#### All Meeting Minutes

##### **Meeting 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Kick-off Meeting | **Date** | May 14, 2013 |
| **Conductor** | Mr.VinhCX | **Time** | 18h-20h |
| **Location** | No 62 Duy Tan Street - Cau Giay district | **Secretary** | HoangPM |
| **Attendees** | Mr.VinhCX, KhanhVD, SonLA, DungNT, HungPm, HoangPM | | |
| **Absent** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Key Point discussed** | | |
| **No** | **Topic** | **Highlight** |
| **1** | Get acquainted | The project team and supervisor get acquainted with each other. |
| **2** | Project discussion | The project team discussed about what’s project does, and what to do (summary). |
| **3** | Project introduction | Brief introduction about project. |
| **4** | System proposal discussions | Guiding of supervisor about system-to-be |
| **5** | Source control | The project team discussed to specify source control management. |
| **6** | Working model | Set up next meetings in schedule, and set up working model |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Plan** | | | |
| **No** | **Action Item(s)** | **Owner** | **Death line** |
| **1** | Create source control repository | KhanhVD | May 14,2013 |
| **2** | Initial project | Project Team | May 15,2013 |
| **3** | Create Project Chatter | KhanhVD | May 15,2013 |
| **4** | Create Task Assignment in week | KhanhVD | May 15,2013 |
| **5** | Technical Research | Project Team | May 16 ,2013 |

##### **Meeting 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Meeting 2 | **Date** | May 21, 2013 |
| **Conductor** | Mr.VinhCX | **Time** | 18h-20h |
| **Location** | No 62 Duy Tan Street - Cau Giay district | **Secretary** | KhanhVD |
| **Attendees** | Mr.VinhCX, KhanhVD, SonLA, DungNT, HungPm, HoangPM | | |
| **Absent** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Key Point discussed** | | |
| **No** | **Topic** | **Highlight** |
| **1** | Project discussion | The project team discussed about what to do (summary). |
| **2** | Review all of result in last week | The Project team review each other result in last week |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Plan** | | | |
| **No** | **Action Item(s)** | **Owner** | **Death line** |
| **1** | Create task assignment in week | KhanhVD | May 21,2013 |
| **2** | Create Report 1 | KhanhVD | May 22,2013 |
| **3** | Task Management Introduction & History | DungNT | May 24,2013 |
| **4** | Initial Ideal of the Group | HungLT | May 24,2013 |
| **5** | Review Existing Products & Systems | SonLA | May 24,2013 |
| **6** | Our solution and Purposes | HoangPM | May 24,2013 |
| **7** | Review and update Report No.1 | KhanhVD | May 25,2013 |

##### **Meeting 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Meeting 3 | **Date** | May 28, 2013 |
| **Conductor** | Mr.VinhCX | **Time** | 18h-20h |
| **Location** | No 62 Duy Tan Street - Cau Giay district | **Secretary** | KhanhVD |
| **Attendees** | Mr.VinhCX, KhanhVD, SonLA, DungNT, HungPm, HoangPM | | |
| **Absent** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Key Point discussed** | | |
| **No** | **Topic** | **Highlight** |
| **1** | Continue discussing about project plan | The project team discussed about some problem of project planning |
| **2** | Review all of result in last week | The Project team review each other result in last week |
| **3** | Task Management on next week | The project team guided each other in task management on next week |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Plan** | | | |
| **No** | **Action Item(s)** | **Owner** | **Death line** |
| **1** | Create task assignment in week | KhanhVD | May 28,2013 |
| **2** | Create report No. 2 | KhanhVD | May 30,2013 |
| **3** | Project Overview | KhanhVD | May 28,2013 |
| **4** | Project Organization | KhanhVD | May 29,2013 |
| **5** | Project Management Plan | KhanhVD | May 30,2013 |
| **6** | Coding Convention | KhanhVD | May 30,2013 |
| **7** | Review and update Report No.2 | KhanhVD | June 4,2013 |
| **8** | Introduction | DungNT | May 29, 2013 |
| **9** | General User Requirement Specification | DungNT | May 31,2013 |
| **10** | External Interface Requirement | DungNT | June 3,2013 |
| **11** | Non-Functional Requirement | HoangPM | May 29, 2013 |
| **12** | List of Functions | HoangPM | June 1, 2013 |
| **13** | Function Description | HoangPM | June 3, 2013 |
| **14** | User Case of User Management Function | HungLT | May 30,2013 |
| **15** | User Case of Message Function | HungLT | June 4,2013 |
| **16** | User Case of Task Management Function | SonLA | June 4, 2013 |

##### **Meeting 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Meeting 4 | **Date** | June 4, 2013 |
| **Conductor** | Mr.VinhCX | **Time** | 18h-21h |
| **Location** | No 62 Duy Tan Street - Cau Giay district | **Secretary** | KhanhVD |
| **Attendees** | Mr.VinhCX, KhanhVD, SonLA, DungNT, HungPm, HoangPM | | |
| **Absent** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Key Point discussed** | | |
| **No** | **Topic** | **Highlight** |
| **1** | Project discussion | The project team discussed about software designing |
| **2** | Review all of result in last week | The Project team review each other result in last week |
| **3** | Task Management on next week | The project team guided each other in task management on next week |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Plan** | | | |
| **No** | **Action Item(s)** | **Owner** | **Death line** |
| **1** | Create task assignment in week | KhanhVD | June 4,2013 |
| **2** | Create Report 3 | KhanhVD | June 4,2013 |
| **3** | System Architecture Design | HoangPM | June 28,2013 |
| **4** | User Interface Design | HungLT, SonLA | June 28,2013 |
| **5** | Identify Tables –Database Design | KhanhVD | June 28,2013 |
| **6** | Entity Relationship Diagram | HoangPM | June 28 ,2013 |
| **7** | Tables Descriptions | HoangPM | June 28,2013 |
| **8** | Coding Interface | DungNT | June 28,2013 |

##### **Meeting 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Meeting 4 | **Date** | June 18, 2013 |
| **Conductor** | Mr.VinhCX | **Time** | 18h-21h |
| **Location** | No 62 Duy Tan Street - Cau Giay district | **Secretary** | HungLT |
| **Attendees** | Mr.VinhCX, KhanhVD, SonLA, DungNT, HungPm, HoangPM | | |
| **Absent** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Key Point discussed** | | |
| **No** | **Topic** | **Highlight** |
| **1** | Review all of result in last week | The Project team review each other result in last week |
| **2** | Task Management on next week | The project team guided each other in task management on next week |
| **3** | Guiding of supervisor | Guiding of supervisor about updating use case, requirement |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Plan** | | | |
| **No** | **Action Item(s)** | **Owner** | **Death line** |
| **1** | User Interface Design | HungLT, SonLA | June 28,2013 |
| **2** | Identify Tables –Database Design | KhanhVD | June 28,2013 |
| **3** | Entity Relationship Diagram | HoangPM | June 28 ,2013 |
| **4** | Tables Descriptions | HoangPM | June 28,2013 |
| **5** | Coding Interface | DungNT, SonLA | June 28,2013 |
| **6** | Update Use case | HungLT | July 8,2013 |
| **7** | Update User Requirement | HungLT | July 8,2013 |
| **8** | Update Tool & Technique description | HoangPM | July 1,2013 |
| **9** | System Coding | KhanhVD | July 30,2013 |
| **10** | Update Report 3 | HungLT | July 10,2013 |

##### **Meeting 6**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Meeting6 | **Date** | July 9, 2013 |
| **Conductor** | Mr.VinhCX | **Time** | 18h-21h |
| **Location** | No 62 Duy Tan Street - Cau Giay district | **Secretary** | HungLT |
| **Attendees** | Mr.VinhCX, KhanhVD, SonLA, DungNT, HungPm, HoangPM | | |
| **Absent** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Key Point discussed** | | |
| **No** | **Topic** | **Highlight** |
| **1** | Project discussion | The project team discussed about software designing |
| **2** | Review all of result in last weeks | The Project team review each other result in last week |
| **3** | Project Screen | Coding team show off all project screen to all member to review and discussed how to make the system better. |
| **4** | Task Management on next week | The project team guided each other in task management on next week. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Plan** | | | |
| **No** | **Action Item(s)** | **Owner** | **Death line** |
| **1** | Create task assignment in week | KhanhVD | July 9,2013 |
| **2** | Create Report 4 | KhanhVD | July 11,2013 |
| **3** | Interface Coding (cont) | DungNT, SonLA | July 30,2013 |
| **4** | System Coding (cont) | KhanhVD | July 30,2013 |
| **5** | Sequence Diagram | HungLT, DungNT, SonLA | July 10,2013 |
| **6** | Design Logo, Button | HoangPM | July 16,2013 |
| **7** | Create Meeting report | HungLT | July 10,2013 |

##### **Meeting 7**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Meeting6 | **Date** | July 16, 2013 |
| **Conductor** | Mr.VinhCX | **Time** | 18h-21h |
| **Location** | No 62 Duy Tan Street - Cau Giay district | **Secretary** | HungLT |
| **Attendees** | Mr.VinhCX, KhanhVD, SonLA, DungNT, HungPm, HoangPM | | |
| **Absent** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Key Point discussed** | | |
| **No** | **Topic** | **Highlight** |
| **1** | Project discussion | The project team discussed about coding plan and some problem in project document. |
| **2** | Review all of result in last weeks | The Project team review each other result in last week |
| **3** | Task Management on next week | The project team guided each other in task management on next week. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Plan** | | | |
| **No** | **Action Item(s)** | **Owner** | **Death line** |
| **1** | Create task assignment in week | KhanhVD | July 9,2013 |
| **2** | Interface Coding (cont) | DungNT, SonLA | July 30,2013 |
| **3** | System Coding (cont) | KhanhVD | July 30,2013 |
| **4** | System Design Description | HungLT | July 20,2013 |
| **5** | Create Test Plan, Test Case | HoangPM | July 30,2013 |
| **6** | Update Report 4 | HungLT | July 18,2013 |

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**

##### **User Management**

* 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 :
* Consumer user is able to send message in Contact screen, he/she touch in user avatar to chose send message function like business admin

##### **Task Management**

* 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.
    - 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**

##### **User Management**

* Business admin 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.

##### **Message and Notification**

* **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.
* Business admin is able to send message from Contact too.

##### **Task Management**

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

#### External Interface Requirements

##### **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

##### **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

##### **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.

##### **Communication Protocol**

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

#### Non-Functional Requirement

##### **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.

##### **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.

##### **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

##### **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

##### **Availability**

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

##### **Security**

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

#### Functional Requirement

##### **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 |

##### **Project Management Function**

|  |  |
| --- | --- |
| FR13 | Create Project |
| FR14 | View Project |
| FR15 | Create Task |
| FR16 | Assign Task Member |
| FR17 | Edit Task Member |
| FR18 | Comment in Task |
| FR29 | Move Task |
| FR20 | Create State |
| FR21 | Delete State |
| FR22 | Rename State |

##### **Message Function**

|  |  |
| --- | --- |
| FR23 | Send Message to a member |
| FR24 | Send Message when a task over due |
| FR25 | Send warning message about over due |
| FR26 | Send Message when making a event |
| FR27 | View user Message |
| FR28 | View system Message |

1. **User Case**
2. **Main Use Case**



1. **Use Case Specification**
2. **Login Function**



|  |  |
| --- | --- |
| **Use Case ID** | **L-001** |
| **Use Case Name** | User registration |
| Date | 5/31/2013 |
| Version | 1.0 |
| Priority | High |
| Author | HungLT |
| Brief Description | This function allows registering a user to use application. |
| Actors | User |
| 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** | **L-002** |
| **Use Case Name** | **Login by Gmail** |
| Date | 5/31/2013 |
| Version | 1.0 |
| Priority | High |
| Author | HungLT |
| Brief Description | This function allows to members access application by Gmail |
| Actors | User |
| 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** | **L-003** |
| **Use Case Name** | **Login by User ID** |
| Date | 5/31/2013 |
| Version | 1.0 |
| Priority | High |
| Author | HungLT |
| Brief Description | This function allows to members access in application by use a id has registered before. |
| Actors | User |
| 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** | **L-004** |
| **Use Case Name** | **User Logout** |
| Version | 1.0 |
| Priority | Normal |
| Date | 6/3/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to members access application by Gmail |
| Actors | User |
| System | Mobile Project Management. |
| Preconditions | Access in “Organization” screen. |
| Post-conditions | Display User Login 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. | |

1. **Project Management Function**



|  |  |
| --- | --- |
| **Use Case ID** | **T-001** |
| **Use Case Name** | **Create Project** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| Brief Description | The application will be make a name to the list for uses to manage their project |
| System | Mobile Project Management |
| Actors | Admin |
| 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** | **View Project** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| System | Mobile Project Management |
| 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** | **Create Task** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| System | Mobile Project Management |
| 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 state board | |
| 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** | **Move task** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| System | Mobile Project Management |
| 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 broad | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-005** |
| **Use Case Name** | **Edit task** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| System | Mobile Project Management |
| 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 information |
| Flow of Events: Edit task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch a task on state board | “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** | **Assign Task** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| System | Mobile Project Management |
| 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 state board | “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** | **Comment in task** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| System | Mobile Project Management |
| 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** | **View my task** |
| Version | 1.0 |
| Priority | High |
| Date | 6/1/2013 |
| Author | SonLA |
| System | Mobile Project Management |
| 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’ user has assigned |
| Flow of Events: View my task | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Access in “Organization” screen |  |  |  | | 2 | Touch button “View my task” | The list of task’ user has  assigned display | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-009** |
| **Use Case Name** | **Create State** |
| Version | 1.2 |
| Priority | High |
| Date | 2/7/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows users create a new state for project |
| Actors | Users |
| Preconditions | Access in “State Board Setting”. |
| Post-conditions | A new state is added. |
| Flow of Events: Create State | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch add “Stateboard” | Display a sub box. |  |  | | 2 | Enter a name. |  | | 3 | Touch “ Save” | A new State is added. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-010** |
| **Use Case Name** | **Rename State** |
| Version | 1.2 |
| Priority | High |
| Date | 7/2/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows users create a new state for project |
| Actors | Users |
| Preconditions | Access in “State Board Setting”. |
| Post-conditions | Rename the chosen state |
| Flow of Events: Rename State | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | User touch a Stateboard that his/ her want to rename | Display a drop list. |  |  | | 2 | Touch “rename”. | Display a sub box. | | 3 | Enter a name. |  |  |  | | 4 | Touch “ Save” | The name of chosen state is  renamed. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **T-011** |
| **Use Case Name** | **Delete State** |
| Version | 1.2 |
| Priority | High |
| Date | 7/2/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows users create a new state for project |
| Actors | Users |
| Preconditions | Access in “State Board Setting”. |
| Post-conditions | The chosen state is deleted. |
| Flow of Events: Delete State | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | User touch a Stateboard that his/ her want to delete | Display a drop list. |  |  | | 2 | Touch “Delete” | Display warning box. | | 3 | Touch “ Yes” | The chosen state(contain all  Information in state) is  deleted. |  |  | |

1. **User Management Function**



|  |  |
| --- | --- |
| **Use Case ID** | **U-001** |
| **Use Case Name** | **Add Member** |
| Version | 1.0 |
| Priority | High |
| Date | 5/29/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | The main function for a group to manage their project. |
| Actors | Admin |
| 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-002** |
| **Use Case Name** | **View Member** |
| Version | 1.0 |
| Priority | High |
| Date | 5/29/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| 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  subbox. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-003** |
| **Use Case Name** | **Change Display Name** |
| Version | 1.2 |
| Priority | High |
| Date | 7/2/2013 |
| Author | HungLT |
| Brief Description | This function allows users to change or set his/ her display name in project. |
| Actors | Users |
| System | Mobile Project Management |
| Preconditions | Access in “Organization” Screen. |
| Post-conditions | Name of user is changed. |
| Flow of Events : Change Display Name | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Profile” icon. | Display User Profile.(In a first time  login, the display name of user is  blank.) |  |  | | 2 | Touch ”Change Display Name” | Display a sub box |  |  | | 3 | Enter a name |  |  |  | | 4 | Touch ”Save” | Name of user is changed or set  successful |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-004** |
| **Use Case Name** | **Change Avatar** |
| Version | 1.2 |
| Priority | High |
| Date | 7/2/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows users to change or set his/ her display avatar. |
| Actors | Users |
| System | Mobile Project Management |
| Preconditions | Access in “Organization” Screen. |
| Post-conditions | Name of user is changed. |
| Flow of Events : Change Avatar | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Profile” icon. | Display User Profile.(In a first time  login, the display avatar of user is  default avatar.) |  |  | | 2 | Touch ”Change Avatar” | Display a sub box |  |  | | 3 | Use brow a image from cell phone |  |  |  | | 4 | Touch ”Save” | Avatar of user is changed. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-005** |
| **Use Case Name** | **Remove Member from Project** |
| Version | 1.0 |
| Priority | High |
| 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 | Admin |
| 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. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-006** |
| **Use Case Name** | **Add Contact** |
| Version | 1.1 |
| Priority | High |
| Date | 30/6/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to members add other member to a list to easy to contact. |
| Actors | User |
| System | Mobile Project Management. |
| Preconditions | Access in “Organization” screen. |
| Post-conditions | A user is added to list. |
| Flow of Events : Add contact | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Contact ” button |  |  |  | | 2 | Touch “Add contact” | Display a sub drop down box. |  |  | | 3 | Enter some characters | Display some simulation of users. |  |  | | 4 | Chose an user in drop down box | A User is added to contact list. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **U-007** |
| **Use Case Name** | **Remove Contact** |
| Version | 1.1 |
| Priority | High |
| Date | 30/6/2013 |
| Author | HungLT |
| Brief Description | This function allows to member remove a member his/her want from added list. |
| Actors | User |
| System | Mobile Project Management. |
| Preconditions | User added to contact list. |
| Post-conditions | A user is removed from list |
| Flow of Events : Remove contact | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Touch “Contact ” button |  |  |  | | 2 | Touch A user in a list | Display a sub box. |  |  | | 3 | Touch “Remove from list” | That user is removed from contact list |  |  | |

1. **Message Management Function**



|  |  |
| --- | --- |
| **Use Case ID** | **M-001** |
| **Use Case Name** | **Send Message to Member** |
| Version | 1.0 |
| Priority | High |
| Date | 5/30/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to communicate between two of members |
| Actors | Users |
| Preconditions | General user access in “Contact” Screen  Business Admin access in “Organization” Screen or “Contact” Screen |
| 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 Over Due Task Message** |
| Version | 1.0 |
| Priority | High |
| Date | 5/31/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to remind users about overdue 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 overdue time | A notification will be sent to all  assigned member and leader. |  |  | |

|  |  |
| --- | --- |
| **Use Case ID** | **M-003** |
| **Use Case Name** | **Send a Message about Warning Over Due Task Message** |
| Version | 1.0 |
| Priority | High |
| Date | 5/31/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to remind users about overdue 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-004** |
| **Use Case Name** | **View System Message** |
| Version | 1.0 |
| Priority | High |
| Date | 7/2/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to user view new message has been sent by system. |
| Actors | User |
| Preconditions | Access in Project list screen or Project Management screen.  A new message has been sent by system. (An exclamation in red color). |
| 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 in drop list.  Red exclamation is disappear. |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **M-005** |
| **Use Case Name** | **View User Message** |
| Version | 1.0 |
| Priority | High |
| Date | 7/2/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to user view new message has been sent by another user. |
| Actors | User |
| Preconditions | Access in “Organization” 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.  (If user touch “Back” button, that  message will be deleted) |  |  | |



|  |  |
| --- | --- |
| **Use Case ID** | **M-006** |
| **Use Case Name** | **Send Message When Making An Event** |
| Version | 1.1 |
| Priority | High |
| Date | 7/3/2013 |
| Author | HungLT |
| System | Mobile Project Management |
| Brief Description | This function allows to user view new message has been sent by another user. |
| Actors | User |
| Preconditions |  |
| Post-conditions | The message has been sent to user |
| Flow of Events **Making Event Message** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Step | Actor Input | System Response |  |  | | 1 | Marking an event.  It contain one in this event list below :   * Add User * Remove User * Assign a Task * Remove Assign Task * Remove from project * Receive a Message from another user | The message has been sent to user that will be received event. |  |  | |

1. **DESIGN & IMPLEMENTATION**
2. **Design Overview**
3. **Objective**

This document describes the technical and UI designs of the system including architectural design and detailed design of common functions. It also includes the design of database model, and overall design of user interface.

The architectural design describes the overall architecture of the system, and the architecture of each main component and subsystem. It introduces the patterns being used, the role of each component and the role of the system in the working environment.

1. **Scope**

The detailed design describes and dynamic structure for each component and function including class diagrams, class explanations, and sequence diagrams of the main use cases. The detailed design uses notations of Java ,C#.Net programming language as language for developing the system.

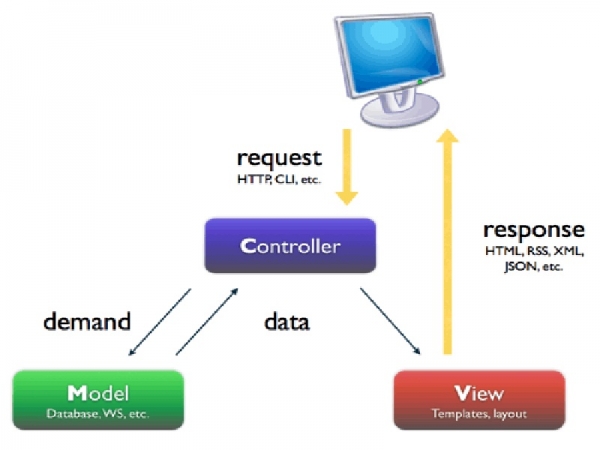
The database design describes the relationship between entities, and details of each entity. It set notations of Window SQL Azure as it is the database server for developing the system.

The user interface design describes the layout of the system, and some design for the screens

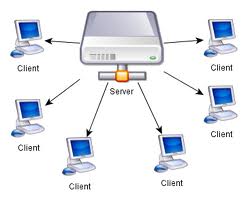
1. **System Architectural Design**

Before choosing system architecture, we made a comparing between MVC model , client –server model and three tier model to decided:

Model–view–controller (MVC) is a software architecture pattern which separates the representation of information from the user's interaction with it. The model consists of application data, business rules, logic, and functions. A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a bar chart for management and a tabular view for accountants. The controller mediates input, converting it to commands for the model or view. The central ideas behind MVC are code reusability and separation of concerns

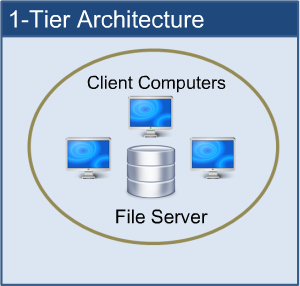


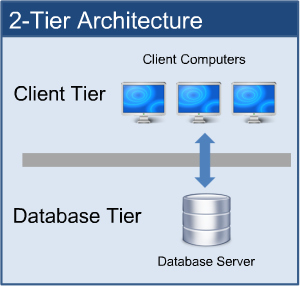
The client–server model is a distributed application structure in computing that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

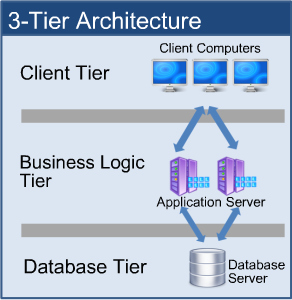


The client–server characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services. The model assigns one of two roles to the computers in a network: Client or server. A server is a computer system that selectively shares its resources; a client is a computer or computer program that initiates contact with a server in order to make use of a resource.

Three-tier model is a client–server model in which presentation, application processing, and data management functions are logically separated.three-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application. A Three-tier architecture is typically composed of a presentation tier, a business or data access tier, and a data tier.







While the concepts of layer and tier are often used interchangeably, one fairly common point of view is that there is indeed a difference. This view holds that a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure.

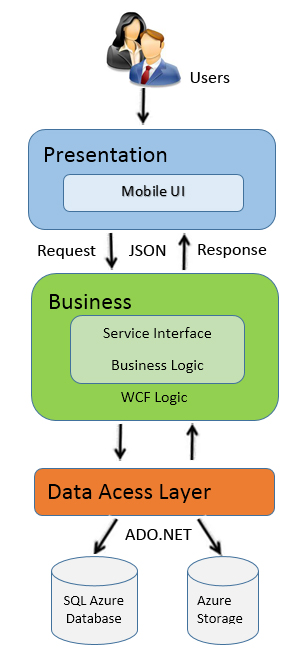
And then , we make a small table when it has used in our system to easier to chose:

|  |  |  |  |
| --- | --- | --- | --- |
| Object | MVC model | Client-Server model | Three tier model |
| Pros. | - Professionalism.  - Less complex in developing  - Easy in maintain, upgrade. | - Can remote access application.  - Integrated modern technology | -Easy to expansion in system scale.  - For big loading, admin adds or removed servers easier. |
| Cons. | - Bulky with small project.  - Spend more time in developing.  - Spend more time in transferring databases from part to part. | - Security and Safety because transferring databases from 2 difference areas. | - Transferring among tiers slower  - Development more complex. |

Because my project is not a very big project, and we only have 4 months to complete it, so we think that bulky is a big problem. We need to use and control our time reasonable.

Besides, in project management, security and safety is a big problem, too.

Based on some reasons above, we chose three –tier model is the main model for our system architecture.



The application will use the Three-Tier architectural model as the main architecture. It makes use of the Internet and JSON protocol to deliver its services to the users.

The main entities of the above diagram will be described as following:

**Presentation Layer**: This is a layer is used to communicate with the user by using Android UI , the main task is that display data and receive data from the users. Simply, this is the interface of the application to communicate with people using the app, receives requests then forwards to the Business Logic. After processing is complete, the application to recognize response from business Logic and displayed to the user that the results of the process. This layer is also responsible for checking the data entered by the user, to control the operation of the user, help user to use the application. This layer contains 4 functions: User, Project management, Messages, Notification functions.

**Business Layer**: Main task is that provides the functionality of the application. This layer contains all the main function of the app, the input and output handler, the data manipulation functions to the request from the Presentation layer, after finishing operations resulting from the Data Tier and pay Presentation on the floor to show the results to the user. This layer uses WCF Service that contains: Service Interface ( Message Manager, Notification, Project management, User management interfaces) and Business Logic ( Message Manager. Notification, Project management, User management business logics ).

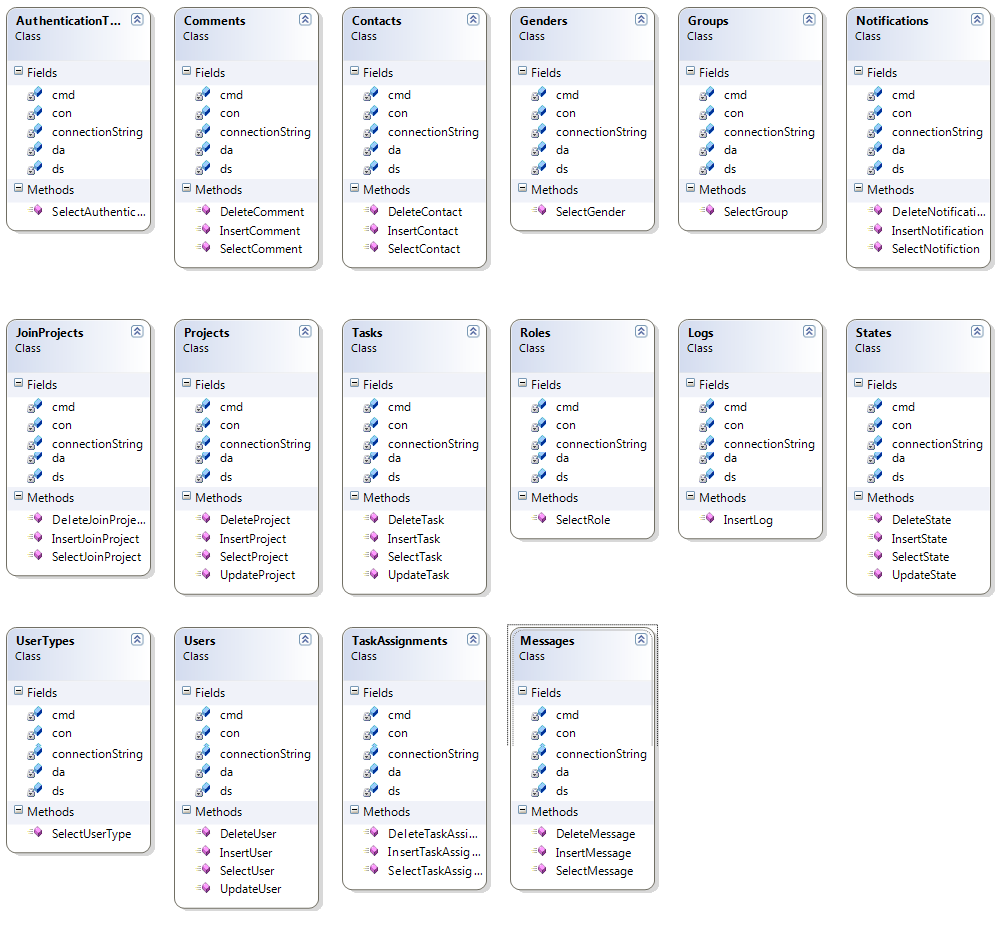
**Data Access Layer**: This is data storage, allowing layers Business Logic can search, extract, update…data then returns results for Business Layer . It contains SQL Azure Database and Azure Storage are the database storage where all the data for application is stored.

1. **Component Diagram**

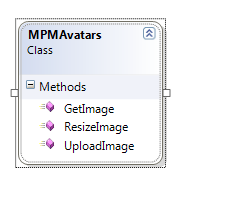


1. **Detail Description of Components**
   * 1. **Class diagram**

#### Data Access Layer

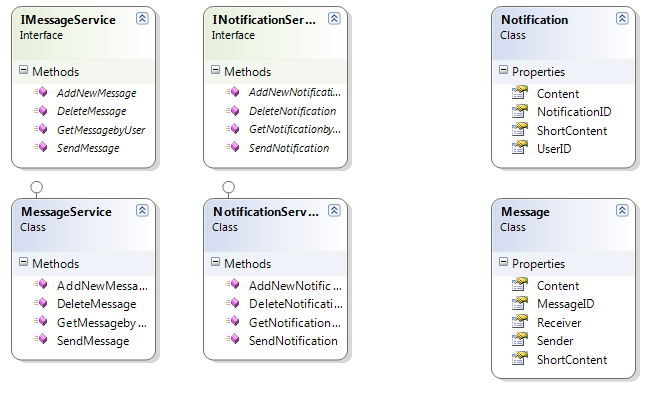
****

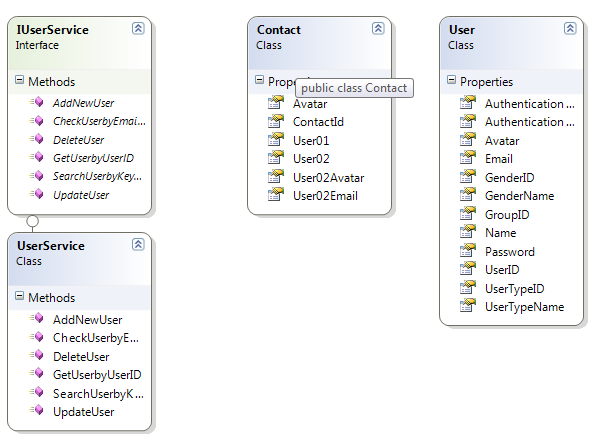
**Figure D-IV-1: AQLDatabase Access Service**

****

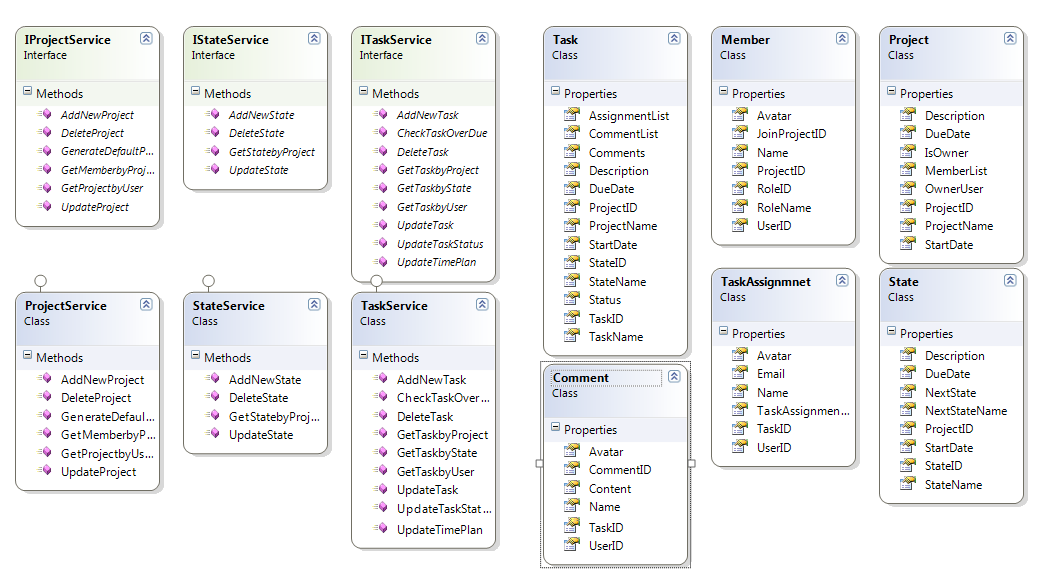
**Figure D-IV-2: Azure Storage Service**

* 1. **Business Logic Layer**

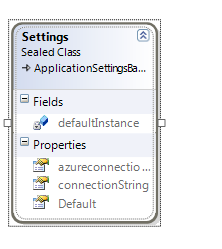
**Figure D-IV-3: Message Notification Service**

****

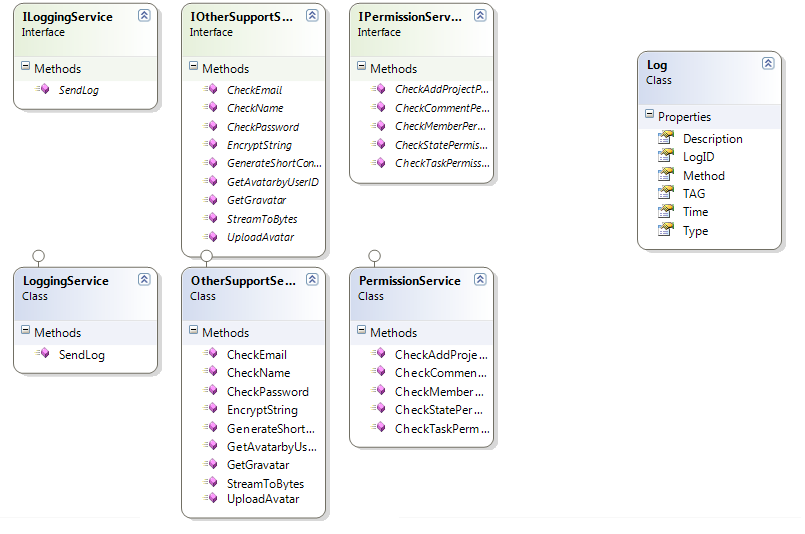
**Figure D-IV-4: User Management Service**

****

**Figure D-IV-5: Project Management Servicce**

****

**Figure D-IV-6: Properties**

****

**Figure D-IV-7: Support Serice**

* + 1. **Class Description**

#### Data Access Layer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Comments** | | | |
| Description | Class to insert into database for activity comment. | | | |
| Extension | Comment, Logs, Log. | | | |
| Implementation | N/A | | | |
| Constructor | N/A | | | |
| Prototype | N/A | | | |
| Source File | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/Comments.cs | | | |
| Namespace | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| Properties | Name | Type | Description | |
| Connection String | string |  | |
| con | SQL Connection |  | |
| cmd | SQL Command |  | |
| da | SQL  DataAdapter |  | |
| con | SQL Connection |  | |
| Methods | Name | Input | Output | Description |
| SelectComment | comment | DataTable | Method to select a comment for delete activity |
| InsertComment | comment | int | Method to insert a comment to database |
| DeleteComment | comment | int | Methof to delete a comment |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **AuthenticationTypes** | | | |
| Description | Class to authentication login | | | |
| Extension | Logs, Log. | | | |
| Implementation | N/A | | | |
| Constructor | N/A | | | |
| Prototype | N/A | | | |
| Source File | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/AuthenticationTypes.cs | | | |
| Namespace | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| Properties | Name | Type | Description | |
| Connection String | string |  | |
| con | SQL Connection |  | |
| cmd | SQL Command |  | |
| da | SQL  DataAdapter |  | |
| con | SQL Connection |  | |
| Methods | Name | Input | Output | Description |
| Select Authentication Type | string | DataTable |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Contacts** | | | |
| Description | Class to control all contact activity | | | |
| Extension | Contact, Logs, Log. | | | |
| Implementation | N/A | | | |
| Constructor | N/A | | | |
| Prototype | N/A | | | |
| Source File | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/Contacts.cs | | | |
| Namespace | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| Properties | Name | Type | Description | |
| Connection String | string |  | |
| con | SQL Connection |  | |
| cmd | SQL Command |  | |
| da | SQL  DataAdapter |  | |
| con | SQL Connection |  | |
| Methods | Name | Input | Output | Description |
| SelectContact | comment | DataTable | Method to select a contact for delete activity |
| InsertContact | comment | int | Method to insert a contact to database |
| DeleteContact | comment | int | Method to delete a contact |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Genders** | | | |
| Description | Class to make difference in display avatar between 2 genders. | | | |
| Extension | Logs, Log. | | | |
| Implementation | N/A | | | |
| Constructor | N/A | | | |
| Prototype | N/A | | | |
| Source File | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/Genders.cs | | | |
| Namespace | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| Properties | Name | Type | Description | |
| Connection String | string |  | |
| con | SQL Connection |  | |
| cmd | SQL Command |  | |
| da | SQL  DataAdapter |  | |
| con | SQL Connection |  | |
| Methods | Name | Input | Output | Description |
| Select Gender | string | DataTable |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Groups** | | | |
| Description | Class to make easier when search a name | | | |
| Extension | Logs, Log. | | | |
| Implementation | N/A | | | |
| Constructor | N/A | | | |
| Prototype | N/A | | | |
| Source File | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/Groups.cs | | | |
| Namespace | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| Properties | Name | Type | Description | |
| Connection String | string |  | |
| con | SQL Connection |  | |
| cmd | SQL Command |  | |
| da | SQL  DataAdapter |  | |
| con | SQL Connection |  | |
| Methods | Name | Input | Output | Description |
| Select  Group | string int  string | DataTable |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **JoinProjects** | | | |
| Description | Class for add project member activity. | | | |
| Extension | Member,Logs, Log. | | | |
| Implementation | N/A | | | |
| Constructor | N/A | | | |
| Prototype | N/A | | | |
| Source File | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/JoinProjects.cs | | | |
| Namespace | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| Properties | Name | Type | Description | |
| Connection String | string |  | |
| con | SQL Connection |  | |
| cmd | SQL Command |  | |
| da | SQL  DataAdapter |  | |
| con | SQL Connection |  | |
| Methods | Name | Input | Output | Description |
| SelectJoin Project | member | DataTable | Method to select a member for remove from project activity. |
| InsertJoin Project | member | int | Method to insert a member to project. |
| DeleteJoin Project | member | int | Method to remover a member form project. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Logs** | | | |
| **Description** |  | | | |
| **Extension** | Log. | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ | | | |
| SQLDatabaseAccessServices/ Logs.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection | string |  | |
| String |
| con | SQL |  | |
| Connection |
| cmd | SQL |  | |
| Command |
| da | SQL |  | |
| DataAdapter |
| con | SQL |  | |
| Connection |
| **Methods** | Name | Input | Output | Description |
| InsertLog | log | int |  |
|
|

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Messages** | | | |
| **Description** | Class to insert into database for activity comment. | | | |
| **Extension** | Message, Logs, Log. | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/Messages.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | SQL Connection |  | |
| cmd | SQL Command |  | |
| da | SQL  DataAdapter |  | |
| con | SQL Connection |  | |
| **Methods** | Name | Input | Output | Description |
| SelectMessage | message | DataTable |  |
| InsertMessage | message | int | Method to insert a message offline to database |
| DeleteMessage | message | int | Methofd to delete a message that the receive has read. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Notifications** | | | |
| **Description** | Class to control all notifications | | | |
| **Extension** | Notification,Logs,Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/Notifications.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | Sql Connection |  | |
| cmd | Sql Command |  | |
| da | Sql DataAdapter |  | |
| ds | DataSet |  | |
| **Methods** | Name | Input | Output | Description |
| Select Notification | notification | DataTable |  |
| Insert Notification | notification | int | Method to save all notification offline or it has not read. |
| Delete Notification | notification | int | Method to delete all notification that it has read. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Projects** | | | |
| **Description** | Class to control all project activity | | | |
| **Extension** | Project,Logs,Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/Projects.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | Sql Connection |  | |
| cmd | Sql Command |  | |
| da | SqlData Adapter |  | |
| ds | DataSet |  | |
| **Methods** | Name | Input | Output | Description |
| SelectProject | project | DataTable |  |
| InsertProject | project | int |  |
| UpdateProject | project | int |  |
| DeleteProject | project | int |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Roles** | | | |
| **Description** |  | | | |
| **Extension** | Logs,Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/  SQLDatabaseAccessServices/Roles.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | Sql Connection |  | |
| cmd | Sql Command |  | |
| da | SqlData Adapter |  | |
| ds | DataSet |  | |
| **Methods** | Name | Input | Output | Description |
| SelectRole | string | DataTable |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **States** | | | |
| **Description** | Class to control all about state . | | | |
| **Extension** | State,Logs,Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/  SQLDatabaseAccessServices/States.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | Sql Connection |  | |
| cmd | Sql Command |  | |
| da | SqlData Adapter |  | |
| ds | DataSet |  | |
| **Methods** | Name | Input | Output | Description |
| SelectState | state | DataTable |  |
| InsertState | state |  |  |
| UpdateState | state |  |  |
| DeleteState | state |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Tasks** | | | |
| **Description** | Class to manage all activity of tasks | | | |
| **Extension** | Task,Logs,Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/  SQLDatabaseAccessServices/Tasks.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | Sql Connection |  | |
| cmd | Sql Command |  | |
| da | SqlData Adapter |  | |
| ds | DataSet |  | |
| **Methods** | Name | Input | Output | Description |
| SelectTask | task | DataTable | Method to select a task for delete task activity |
| InsertTask | task | int | Method to add a task into database |
| UpdateTask | task | int | Method to update into database when changing a task |
| DeleteTask | task | int |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Users** | | | |
| **Description** | Class to manage users | | | |
| **Extension** | User,Logs,Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/  SQLDatabaseAccessServices/Users.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | Sql Connection |  | |
| cmd | Sql Command |  | |
| da | SqlData Adapter |  | |
| ds | DataSet |  | |
| **Methods** | Name | Input | Output | Description |
| SelectUser | user | DataTable |  |
| InsertUser | user | int | Method to add a user to project |
| UpdateUser | user | int |  |
| DeleteUser | user | int | Method to delete a user from a project |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **UserTypes** | | | |
| **Description** | Class to define users type. | | | |
| **Extension** | UserType,Logs,Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ SQLDatabaseAccessServices/UserTypes.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.SQLDatabaseAccessServices | | | |
| **Properties** | Name | Type | Description | |
| Connection String | string |  | |
| con | Sql Connection |  | |
| cmd | Sql Command |  | |
| da | SqlData Adapter |  | |
| ds | DataSet |  | |
| **Methods** | Name | Input | Output | Description |
| SelectUserType | string | DataTable |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **MPMAvatars** | | | |
| **Description** | Class for display avatar. | | | |
| **Extension** | Member,Logs, Log. | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ AzureStorageServices/MPMAvatars.cs | | | |
| **Namespace** | WCFServices.DataAccessLayer.AzureStorageServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
| GetImage | string | Image |  |
| ResizeImage | Image int | Image | Method to resize avatar has been upload smaller |
| UploadImage | string  string  Image | string | Method to upload avatar |

#### Business Logic Layer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **UserService** | | | |
| **Description** |  | | | |
| **Extension** | User | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | IUserService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ BusinessLogicLayer/UserManagementServices/UserService.svc.cs | | | |
| **Namespace** | WCFServices.BusinessLogicLayer.UserManagementServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
| AddNewUser | user | int |  |
| DeleteUser | int | Int |  |
| UploadUser | user | int |  |
| GetUserby UserID | int | user | Method to login by user ID |
| CheckUserby EmailPassword | string  string | int | Method to login by email. |
| SearchUser byKeyWord | string | List<User> | Method to display group of name user when search |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **PermissionService** | | | |
| **Description** | Class to check permission in some activity. | | | |
| **Extension** |  | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | IPermissionService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ | | | |
| BusinessLogicLayer/SupportServices/PermissionService.svc.cs | | | |
| **Namespace** | WCFServices.BusinessLogicLayer.SupportServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
| CheckAddProject Permisson | int | int | Method to check permission for add project. |
|
| CheckState Permission | int | int | Method to check permission in all states activity |
| int |
| CheckTask | int | int | Method to check permission in all tasks activity |
| int |
| int |
| CheckMember Permission | int | int |  |
| int |
| CheckComment Permission | int | int |  |
| int |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **LoggingService** | | | |
| **Description** | Class to control log services | | | |
| **Extension** | Log | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | ILoggingService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ | | | |
| BusinessLogicLayer/SupportServices/LoggingService.svc.cs | | | |
| **Namespace** | WCFServices.BusinessLogicLayer.SupportServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
| SendLog | log | void |  |
|

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | OtherSupportService | | | |
| **Description** | Class include some support sirvice | | | |
| **Extension** | Stream | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | IOtherSupportService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/BusinessLogicLayer/ OtherSupportService.svc.cs | | | |
|
| **Namespace** | WCFServices.BusinessLogicLayer.SupportServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  | |
|  | Name | Input | Output | Description |
| **Methods** | EncryptString | string | string | Method to encrypt password to MD5 |
|
| GenerateShort Content | string | string |  |
| GetAvatarby UserID | int | string | Method to display Gmail avatar |
| GetGravatar | Stream | string | Method to display avatar upload by user |
| UploadAvatar | Stream | string | Method to upload avatar |
| StreamToBytes | Stream | byte[] |  |
| CheckName | string | int | Method to valiadate Name |
| CheckEmail | string | int | Method to validate Email |
| CheckPassword | string | int | Method to validate Password |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **StateService** | | | |
| **Description** | Class include all state sirvice | | | |
| **Extension** | State | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | IStateService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ BusinessLogicLayer/ProjectManagementServices/ProjectService.svc.cs | | | |
| **Namespace** | WCFServices.BusinessLogicLayer.ProjectManagementServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  |  | |
| **Methods** | Name | Input | Output | Description |
| AddNewState | state | int | Method to add a new state to database |
| DeleteState | int | int | Method to delete a state |
| UpdateState | state | int | Method to update state when it change |
| GetState byProject | int | List<State> | Method to list all state of project that use join in |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Class** | **ProjectService** | | | | |
| **Description** | Class include all project service | | | | |
| **Extension** | Project | | | | |
| **Implementation** | N/A | | | | |
| **Constructor** | N/A | | | | |
| **Prototype** | IProjectService | | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/ BusinessLogicLayer/ProjectManagementServices/ProjectService.svc.cs | | | | |
| BusinessLogicLayer/ProjectManagementServices/ProjectService.svc.cs | | | | |
| **Namespace** |  |  |  |  | |
| **Properties** | Name | Type | Description | | |
|  |  |  | | |
| **Methods** | Name | Input | Output | | Description |
| AddNewProject | project | int | | Method to add a new project to database |
| Delete NewProject | int | int | | Method to delete project from database |
| UpdateProject | project | int | | Method to update project when it change |
| GetProjectbyUser | int | List<Project> | | Method to load project that user join in |
| Generate DefaultProject | int | int | |  |
| GetMember byProject | int | List<Member> | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **TaskService** | | | |
| **Description** | Class contain all task service | | | |
| **Extension** | Task,List<Task> | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | ITaskService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/BusinessLogicLayer/ ProjectManagementServices/TaskService.svc.cs | | | |
| **Namespace** | WCFServices.BusinessLogicLayer.ProjectManagementServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  |  | |
| **Methods** | Name | Input | Output | Description |
| AddNewTask | task | int | Method to add a new Task to database |
| DeleteTask | int | int | Method to delete a Task |
| UpdateTask | task | int | Method to update Task when it change |
| GetTaskbyUser | int | List<State> | Method to list all Task of Project that use join in |
| CheckTask OverDue | int | void | Method to check Task overdue or not |
| GetTask byProject | int | List<State> | Method to get all Task of Project |
| GetTask byState | int | List<State> | Method to get all Task of State |
| Update TaskStatus | List<Task> | List<Task> | Method to controll status of Task |
| Update TimePlan | int,DateTime, DateTime | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **NotificationService** | | | |
| **Description** | Class include some Notification service | | | |
| **Extension** | Notification | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | INotificationService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/BusinessLogicLayer /MessageNotificationServices/NotificationService.svc.cs | | | |
| **Namespace** | WCFServices.BusinessLogicLayer.MessageNotificationServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  |  | |
| **Methods** | Name | Input | Output | Description |
| Send Notification | notification | int | Method to add a send new notification |
| AddNew Notification | notification | int | Method to add new notification to send |
| GetNotification byUser | int | List<Notification> | Method to list all offline notification |
| Delete Notification | int | int | Method to delete notification has read |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **MessageService** | | | |
| **Description** | Class include all message service | | | |
| **Extension** | Message | | | |
| **Implementation** | N/A | | | |
| **Constructor** | N/A | | | |
| **Prototype** | IMessageService | | | |
| **Source File** | ./MPMServerSide/WCFServices/DataAccessLayer/BusinessLogicLayer /MessageNotificationServices/MessageService.svc.cs | | | |
| **Namespace** | WCFServices.BusinessLogicLayer.MessageNotificationServices | | | |
| **Properties** | Name | Type | Description | |
|  |  |  |  | |
| **Methods** | Name | Input | Output | Description |
| Send Message | message | int | Method to add a send new message |
| AddNew Message | message | int | Method to add new message to send |
| GetMessage byUser | int | List<Message> | Method to list all offline message |
| Delete Message | int | int | Method to delete message has read |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mUser.java** | | | |
| **Description** | Class contain all model of User | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mUser | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mUser.java | | | |
| **Namespace** |  | | | |
| **Properties** | Name | Type | Description | |
| userId | int |  | |
| name | String |  | |
| groupId | int |  | |
| email | String |  | |
| password | String |  | |
| avatar | String |  | |
| userTypeId | String |  | |
| userTypeName |  |  | |
| genderId | String |  | |
| genderName | String |  | |
| Authentication TypeId | String |  | |
| Authentication TypeName | String |  | |
| **Methods** | Name | Input | Output | Description |
| getUserID |  | int |  |
| setUserID | int | void |  |
| getName |  | String |  |
| setName | String | void |  |
| getGroupID |  | int |  |
| setGroupID | int | void |  |
| getEmail |  | String |  |
| setEmail | String | void |  |
| getPassword |  | String |  |
| setPassword | String | void |  |
| getAvatar |  | String |  |
| setAvatar | String | void |  |
| getUserTypeID |  | String |  |
| setUserTypeID | String | void |  |
| getUser TypeName |  | String |  |
| setUser TypeName | String | void |  |
| getGenderID |  | String |  |
| setGenderID | String | void |  |
| getGenderName |  | String |  |
| setGenderName | String | void |  |
| getAuthentication TypeID |  | String |  |
| setAuthentication TypeID | String | void |  |
| etAuthentication TypeName |  | String |  |
| setAuthentication TypeName | String | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mTaskAssignment.java** | | | |
| **Description** | Class contain all Task Assignment model | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mTaskAssignment | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mTaskAssignment.java | | | |
| **Namespace** |  | | | |
| **Properties** | Name | Type | Description | |
| taskAssignmentId | int |  | |
| userId | int |  | |
| name | String |  | |
| email | String |  | |
| avatar | String |  | |
| taskId | int |  | |
| **Methods** | Name | Input | Output | Description |
| Get TaskAssignmentId |  | int |  |
| Set TaskAssignmentID | int | void |  |
| getUserId |  | int |  |
| setUserID | int | void |  |
| getName |  | String |  |
| setName | String | void |  |
| getEmail |  | String |  |
| setEmail | String | void |  |
| getAvatar |  | String |  |
| setAvatar | String | void |  |
| getTaskId |  | int |  |
| setTaskID | int | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mTask.java** | | | |
| **Description** | Class contain all Task model | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mTask | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mTask.java | | | |
| **Namespace** |  | | | |
| **Properties** | Name | Type | Description | |
| taskId | int |  | |
| taskName | String |  | |
| description | String |  | |
| stateId | int |  | |
| stateName | String |  | |
| projectId | int |  | |
| projectName | String |  | |
| comments | int |  | |
| status | String |  | |
| Assignment List | List<mTask Assignment> |  | |
| Comment List | List<m  Comment> |  | |
| startDate | Date |  | |
| dueDate | Date |  | |
| **Methods** | Name | Input | Output | Description |
| getTaskID |  | int |  |
| setTaskID | int | void |  |
| getTaskName |  | String |  |
| setTaskName | String | void |  |
| getDescription |  | String |  |
| setDescription | String | void |  |
| getStateID |  | int |  |
| setStateID | int | void |  |
| getStateName |  | String |  |
| setStateName | String | void |  |
| getProjectID |  | int |  |
| setProjectID | int | void |  |
| getProjectName |  | String |  |
| setProjectName | String | void |  |
| getComments |  | int |  |
| setComments | int | void |  |
| getStatus |  | String |  |
| setStatus | String | void |  |
| getTaskAssignment |  | List<mTask Assignment> |  |
| setTaskAssignment | List<mTask Assignment> | void |  |
| getCommentList |  | List<mComment> |  |
| setCommentList | List<m Comment> | void |  |
| getStartDate |  | Date |  |
| setStartDate | Date | void |  |
| getDueDate |  | Date |  |
| setDueDate | Date | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mState.java** | | | |
| **Description** | Class contain all State model | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mState | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mState.java | | | |
| **Namespace** |  | | | |
| **Properties** | Name | Type | Description | |
| stateId | int |  | |
| stateName | String |  | |
| description | String |  | |
| projectId | int |  | |
| nextState | int |  | |
| nextStateName | String |  | |
| startDate | String |  | |
| dueDate | int |  | |
| **Methods** | Name | Input | Output | Description |
| getStateId |  | int |  |
| setStateID | int | void |  |
| getStateName |  | String |  |
| setStateName | String | void |  |
| getDescription |  | String |  |
| setDescription | String | void |  |
| getProjectId |  | int |  |
| setProjectID | int | void |  |
| nextState |  | int |  |
| setNextState | int | void |  |
| getNextStateName |  | String |  |
| setNextStateName | String | void |  |
| getStartDate |  | Date |  |
| setStartDate | Date | void |  |
| getDueDate |  | Date |  |
| setDueDate | Date | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mProject.java** | | | |
| **Description** | Class contain all Project model | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mProject | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mProject.java | | | |
| **Namespace** |  | | | |
| **Properties** | Name | Type | Description | |
| projectId | int |  | |
| projectName | String |  | |
| description | String |  | |
| ownerUser | int |  | |
| isOwner | int |  | |
| memberList | List<m Member> |  | |
| startDate | String |  | |
| dueDate | int |  | |
| **Methods** | Name | Input | Output | Description |
| getProjectID |  | int |  |
| setProjectID | int | void |  |
| getProjectName |  | String |  |
| setProjectName | String | void |  |
| getDescription |  | String |  |
| setDescription | String | void |  |
| getOwnerUser |  | int |  |
| setOwnerUser | int | void |  |
| getIsOwner |  | int |  |
| setIsOwner | int | void |  |
| getMemberList |  | List<m Member> |  |
| setMemberList | List<m Member> | void |  |
| getStartDate |  | Date |  |
| setStartDate | Date | void |  |
| getDueDate |  | Date |  |
| setDueDate | Date | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mNotification.java** | | | |
| **Description** | Class contain all Notification model | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mProject | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mNotification.java | | | |
| **Namespace** |  | | | |
| **Properties** | Name | Type | Description | |
| notificationId | int |  | |
| userId | int |  | |
| content | String |  | |
| shortContent | String |  | |
| **Methods** | Name | Input | Output | Description |
| getNotificationID |  | int |  |
| etNotificationID | int | void |  |
| getUserID |  | int |  |
| setUserID | int | void |  |
| getContent |  | String |  |
| setContent | String | void |  |
| getShortContent |  | String |  |
| setShortContent | String | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mMessage.java** | | | |
| **Description** |  | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mMessage | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mMessage.java | | | |
| **Namespace** | N/A | | | |
| **Properties** | Name | Type | Description | |
| messageId | int |  | |
| sender | int |  | |
| receiver | int |  | |
| content | string |  | |
| shortContent | string |  | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
| getMessageID |  | int |  |
| setMessageID | int | void |  |
| getSender |  | int |  |
| setSender | int | void |  |
| getReceiver |  | int |  |
| setReceiver | int | void |  |
| getContent |  | string |  |
| setContent | string | void |  |
| getShortContent |  | string |  |
| setShortContent | string | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mComment.java** | | | |
| **Description** |  | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mComment | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mComment.java | | | |
| **Namespace** | N/A | | | |
| **Properties** | Name | Type | Description | |
| commentId | int |  | |
| content | int |  | |
| userID | int |  | |
| name | string |  | |
| avatar | string |  | |
| taskId | int |  | |
| **Methods** | Name | Input | Output | Description |
| getCommentId |  | int |  |
| setCommentID | int | void |  |
| getContent |  | string |  |
| setContent | string | void |  |
| getuserId |  | int |  |
| setuserID | int | void |  |
| getName |  | string |  |
| setName | string | void |  |
| getAvatar |  | string |  |
| setAvatar | string | void |  |
| getTaskId |  | int |  |
| setTaskId | int | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mContact.java** | | | |
| **Description** |  | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mContact | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mContact.java | | | |
| **Namespace** | N/A | | | |
| **Properties** | Name | Type | Description | |
| contactId | int |  | |
| user01 | int |  | |
| user02 | int |  | |
| user02Email | string |  | |
| user02Name | string |  | |
| user02Avatar | string |  | |
| **Methods** | Name | Input | Output | Description |
| getContactID |  | int |  |
| setContactID | int | void |  |
| getUser01 |  | int |  |
| setUser01 | int | void |  |
| getUser02 |  | int |  |
| setUser02 | int | void |  |
| getUser02Email |  | string |  |
| setUser02Email | string | void |  |
| getUser02Name |  | string |  |
| setUser02Name | string | void |  |
| getUser02Avatar |  | string |  |
| setUser02Avatar | string | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mLog.java** | | | |
| **Description** |  | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mLog | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mLog.java | | | |
| **Namespace** | N/A | | | |
| **Properties** | Name | Type | Description | |
| logId | int |  | |
| time | Date |  | |
| tag | string |  | |
| method | string |  | |
| type | string |  | |
| description | string |  | |
| **Methods** | Name | Input | Output | Description |
| getLogId |  | int |  |
| setLogID | int | void |  |
| getTime |  | Date |  |
| setTime | Date | void |  |
| getTAG |  | string |  |
| setTAG | string | void |  |
| getMethod |  | string |  |
| setMethod | string | void |  |
| getType |  | string |  |
| setType | string | void |  |
| getDescription |  | string |  |
| setDescription | string | void |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **mMember.java** | | | |
| **Description** |  | | | |
| **Extension** | Observable | | | |
| **Implementation** | Serializable | | | |
| **Constructor** | mMember | | | |
| **Prototype** | N/A | | | |
| **Source File** | ./MPMClientSide\_Final/src/fsg/mpmclientside/models/mMember.java | | | |
| **Namespace** | N/A | | | |
| **Properties** | Name | Type | Description | |
| joinProjectId | int |  | |
| projectId | int |  | |
| name | int |  | |
| avatar | int |  | |
| roleId | string |  | |
| roleName | string |  | |
| **Methods** | Name | Input | Output | Description |
| getJoinProjectID |  | int |  |
| setJoinProjectID | int | void |  |
| getProjectID |  | int |  |
| setProjectID | int | void |  |
| getUserID |  | int |  |
| setUserID | int | void |  |
| getName |  | int |  |
| setName | int | void |  |
| getAvatar |  | int |  |
| setAvatar | int | void |  |
| getRoleID |  | string |  |
| setRoleID | string | void |  |
| getRoleName |  | string |  |
| setRoleName | string | void |  |

1. **Sequence Diagram**
2. **Login Screen**



**Figure D-VI-1: Login Screen**

1. **Sign up Screen**



**Figure D-VI-2: Sign Up Screen**

1. **Add New Project Screen**



**Figure D-VI-3: Add New Project Screen**

1. **Project Member Screen**



**Figure D-VI-4: Project Member Screen**

1. **Add New Member Dialog**



**Figure D-VI-5: Add New Member Screen**

1. **Member Detail Dialog**



**Figure D-VI-6: Member Detail Dialog**

1. **Send Message Screen**



**Figure D-VI-7: Send Message Screen**

1. **Notification Screen**



**Figure D-VI-8: Notification Screen**

1. **Add Task**



**Figure D-VI-9: Add Task Screen**

1. **Edit Task**



**Figure D-VI-10: Edit Task Screen**

1. **Assign Task Screen**



**Figure D-VI-11: Assign Task Screen**

1. **State Board Setting**



**Figure D-VI-12: State Board Screen**

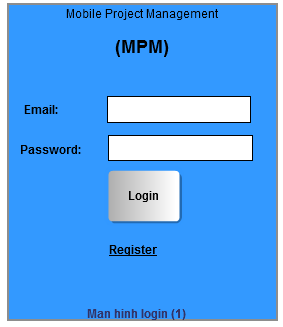
1. **Add State Board Screen**

 **Figure D-VI-13: Add State Board Screen**

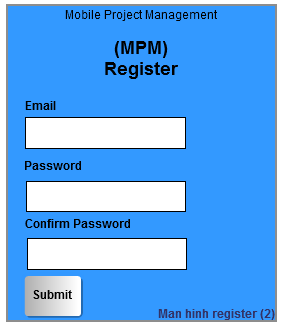
1. **User Interface Design**

Refer to Description of System Interface\_Hoangpm01336\_16thang7 folder HoangPm

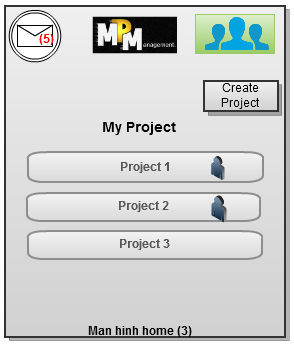
**Login Screen**



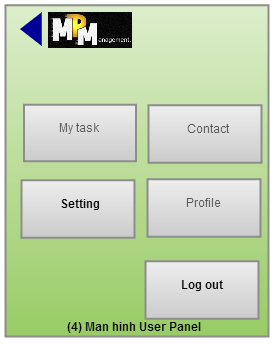
**Register Screen**



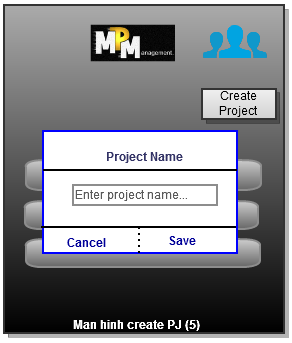
**Home Screen**



**User Screen**



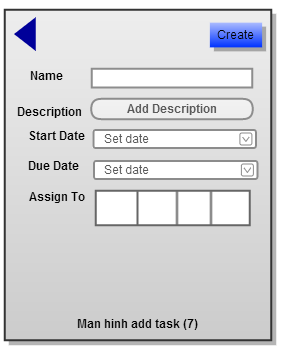
**Create Project Screen**



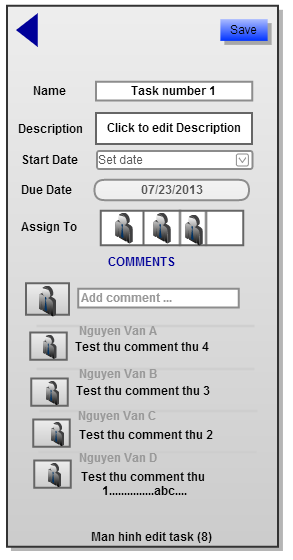
**StateBoard Screen**



**Add Task Screen**



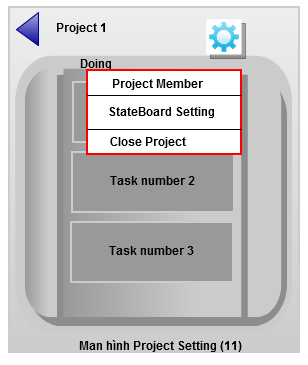
**Edit Task Screen**



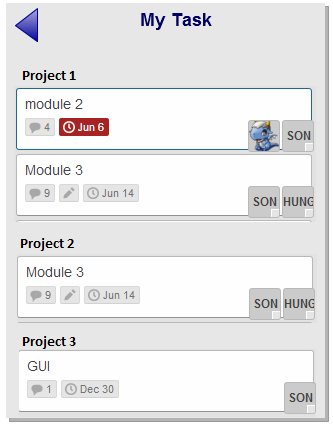
**Assign Task Screen**



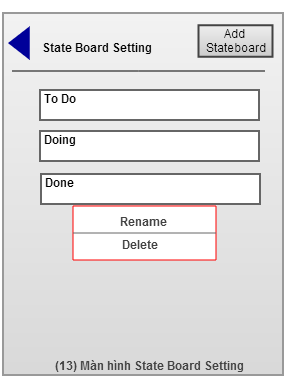
**Project Setting Screen**



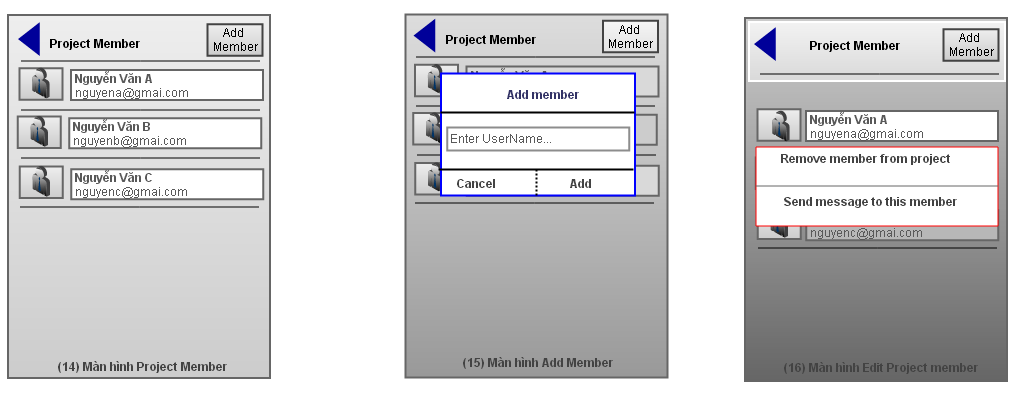
**My task Screen**



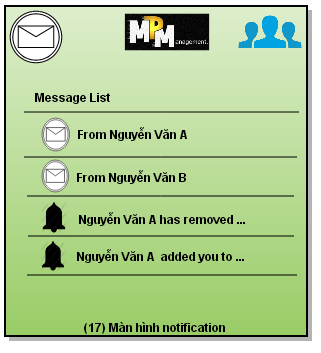
**State board Setting**

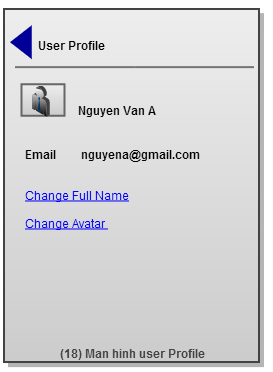


**Project member screen**



**Notification screen**

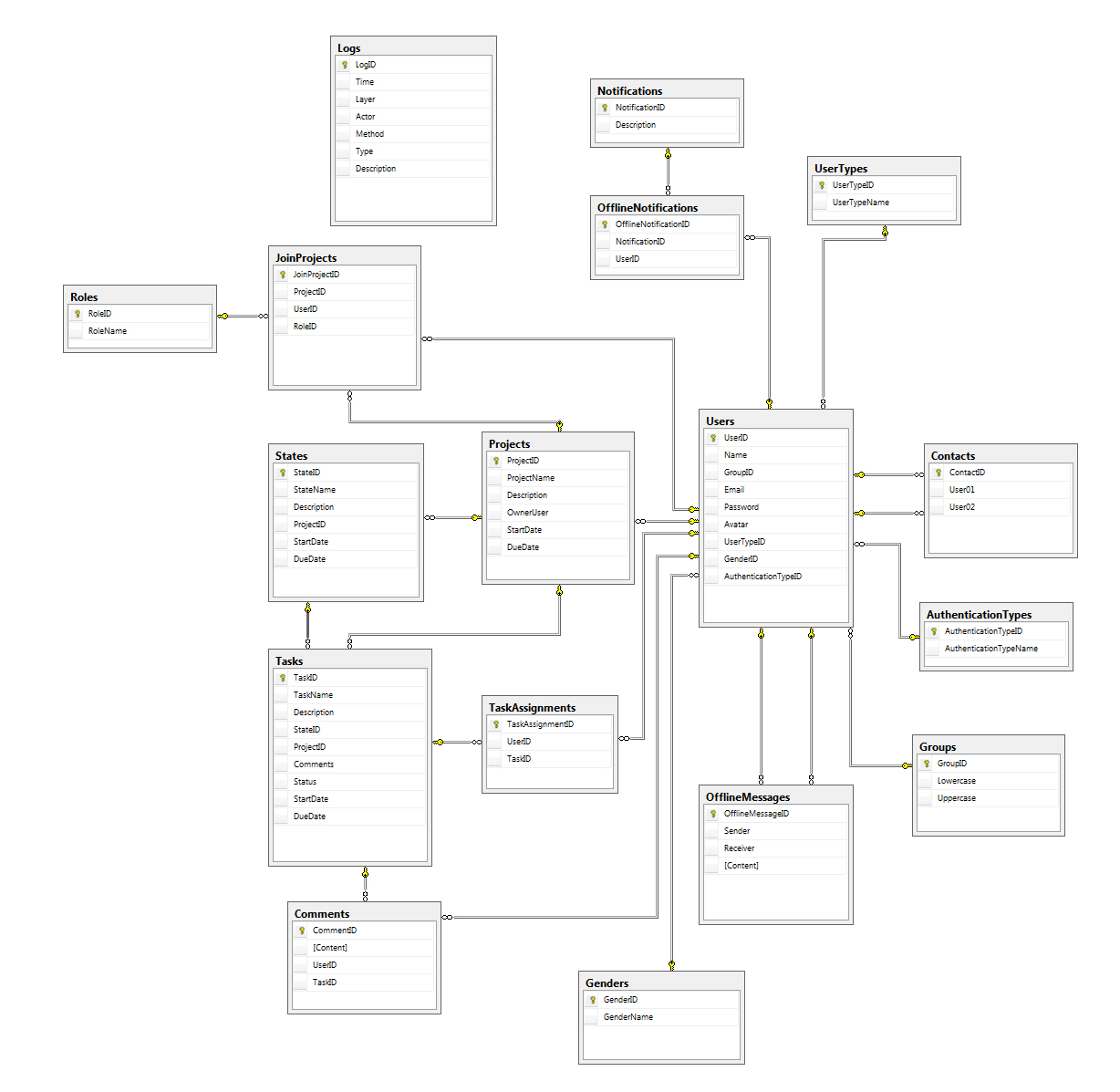


**User profile screen**

**Send message screen**



1. **Database Design**
2. **Entity Relationship Diagram (ERD)**



1. **Table Description**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. **Table: AuthenticationTypes** | | | | | | | | | | | |
| **No** | **Attribute** | | | | | **Type** | | | | **Foreign key** | |
| 1 | AuthenticationTypeID | | | | | nvarchar(5) | | | |  | |
| 2 | AuthenticationTypeName | | | | | nvarchar(15) | | | |  | |
|  |  | | | | |  | | | |  | |
|  |  | | | | |  | | | |  | |
| 1. **Table: Comments** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | CommentID | | | | int | | | | |  | |
| 2 | Content | | | | nvarchar(100) | | | | |  | |
| 3 | UserID | | | | int | | | | | Users(UserID) | |
| 4 | TaskID | | | | int | | | | | Tasks(TaskID) | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: Contacts** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | ContactID | | | | int | | | | |  | |
| 2 | User01 | | | | int | | | | | Users(UserID) | |
| 3 | User02 | | | | int | | | | | Users(UserID) | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: JoinProjects** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | JoinProjectID | | | | int | | | | |  | |
| 2 | ProjectID | | | | int | | | | | Projects(ProjectID) | |
| 3 | UserID | | | | int | | | | | Users(UserID) | |
| 4 | RoleID | | | | nvarchar(5) | | | | | Roles(RoleID) | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: Projects** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | ProjectID | | | | int | | | | |  | |
| 2 | ProjectName | | | | nvarchar(50) | | | | |  | |
| 3 | Description | | | | nvarchar(150) | | | | |  | |
| 4 | OwnerUserID | | | | int | | | | | Users(UserID) | |
| 5 | StartDate | | | | datetime | | | | |  | |
| 6 | DueDate | | | | datetime | | | | |  | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: Roles** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | RoleID | | | | nvarchar(5) | | | | |  | |
| 2 | RoleName | | | | nvarchar(15) | | | | |  | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: States** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | StateID | | | | int | | | | |  | |
| 2 | StateName | | | | nvarchar(15) | | | | |  | |
| 3 | Description | | | | nvarchar(150) | | | | |  | |
| 4 | ProjectID | | | | int | | | | | Projects(ProjectID) | |
| 5 | StartDate | | | | datetime | | | | |  | |
| 6 | DueDate | | | | datetime | | | | |  | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| **Table: TaskAssignments** | | | | | | | | | | | |
| **No** | **Attribute** | | | | | **Type** | | | | **Foreign key** | |
| 1 | TaskAssignmentID | | | | | int | | | |  | |
| 2 | UserID | | | | | int | | | | Users(UserID) | |
| 3 | TaskID | | | | | int | | | | Tasks(TaskID) | |
|  |  | | | | |  | | | |  | |
|  | |  | | | | | | | | |
| 1. **Table: Tasks** | | | | | | | | | | | |
| **No** | **Attribute** | | | | | **Type** | | | | **Foreign key** | |
| 1 | TaskID | | | | | int | | | |  | |
| 2 | TaskName | | | | | nvarchar(50) | | | |  | |
| 3 | Description | | | | | nvarchar(150) | | | |  | |
| 4 | StateID | | | | | int | | | | States(StateID) | |
| 5 | ProjectID | | | | | int | | | | Projects(ProjectID) | |
| 6 | Comments | | | | | int | | | |  | |
| 7 | Status | | | | | nvarchar(10) | | | |  | |
| 8 | StartDate | | | | | datetime | | | |  | |
| 9 | DueDate | | | | | datetime | | | |  | |
|  |  | | | | |  | | | |  | |
| **Table Users** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | UserID | | | | int | | | | |  | |
| 2 | Name | | | | nvarchar(50) | | | | |  | |
| 3 | GroupID | | | | int | | | | | Groups(GroupID) | |
| 4 | Email | | | | nvarchar(50) | | | | |  | |
| 5 | Password | | | | nvarchar(max) | | | | |  | |
| 6 | Avatar | | | | nvarchar(max) | | | | |  | |
| 7 | UserTypeID | | | | nvarchar(5) | | | | | UserTypes(UserTypeID) | |
| 8 | AuthenticationTypeID | | | | nvarchar(5) | | | | | AuthenticationTypes(AuthenticationTypeID) | |
| 9 | Gender | | | | nvarchar(5) | | | | | Genders(GenderID) | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: UserTypes** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | UserTypeID | | | | nvarchar(5) | | | | |  | |
| 2 | UserTypeName | | | | nvarchar(15) | | | | |  | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: Notifications** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | NotifficationID | | | | nvarchar(5) | | | | |  | |
| 2 | Descriptions | | | | nvarchar(150) | | | | |  | |
|  |  | | | |  | | | | |  | |
| 1. **Table: Logs** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | | **Foreign key** | |
| 1 | LogID | | | | int | | | | |  | |
| 2 | Time | | | | datetime | | | | |  | |
| 3 | Layer | | | | nvarchar(30) | | | | |  | |
| 4 | Actor | | | | nvarchar(15) | | | | |  | |
| 5 | Method | | | | nvarchar(15) | | | | |  | |
| 6 | Type | | | | nvarchar(10) | | | | |  | |
| 7 | Description | | | | nvarchar(max) | | | | |  | |
|  |  | | | |  | | | | |  | |
|  |  | | | |  | | | | |  | |
| **Table: Groups** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | **Foreign key** | | |
| 1 | GroupID | | | | int | | | |  | | |
| 2 | Lowercase | | | | char(1) | | | |  | | |
| 3 | Uppercase | | | | char(1) | | | |  | | |
|  |  | | | |  | | | |  | | |
| 1. **Table: OfflineNotifications** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | **Foreign key** | | |
| 1 | OfflineNotificationID | | | | int | | | |  | | |
| 2 | NotificationID | | | | nvarchar(5) | | | | Notifications(NotificationID) | | |
| 3 | UserID | | | | int | | | | Users(UserID) | | |
|  |  | | | |  | | | |  | | |
| 1. **Table: Offline Messages** | | | | | | | | | | | |
| **No** | **Attribute** | | | | **Type** | | | | **Foreign key** | | |
| 1 | OfflineMessageID | | | | int | | | |  | | |
| 2 | Sender | | | | int | | | | Users(UserID) | | |
| 3 | Receiver | | | | int | | | | Users(UserID) | | |
| 4 | Content | | | | nvarchar(150) | | | |  | | |
|  |  | | | |  | | | |  | | |
| 1. **Table: Genders** | | | | | | | | | | | |
| **No** | **Attribute** | | **Type** | | | | **Foreign key** | | | | |
| 1 | GenderID | | nvarchar(5) | | | |  | | | | |
| 2 | GenderName | | nvarchar(10) | | | |  | | | | |
|  |  | |  | | | |  | | | | |
| **Table: AuthenticationTypes** | | | | | | | | | | | |
| **No** | **Attribute** | | | **Type** | | | | **Foreign key** | | | |
| 1 | AuthenticationTypeID | | | nvarchar(5) | | | |  | | | |
| 2 | AuthenticationTypeName | | | nvarchar(15) | | | |  | | | |

1. **TEST DOCUMENTATION**
2. **Test Plan**

### Features to be tested

Here is the list of functions of the system that have to be tested in testing scope:

* 1. Registration function

1. Log in/ log out function
2. User management function
3. Roles function
4. Task management function
5. Project management function
6. State management function
7. Notification management function
8. Message management function
9. Comment function
10. All Screens

### Features not to be tested

Here is the list of the features that do not have to be tested in testing scope.

### Performance

We are trying to ensure the system functions operates correctly first. The performance is not considered in this case.

### User Acceptance Testing

This application will not perform User Acceptance Testing. The purpose of these tests is to confirm that the system is developed according to the specified user requirement.

3. **Test Case**

### Registration function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Register** | | | | | | | |
| Step:   * Open Management Console * From Login Screen tap on Register | | | | | | | |
| MC1.1 | Registered  successfully | 1. Input email and password correctly  2. Tap [Submit] button | 2. Login screen opened |  |  |  |  |
| MC1.2 | Registered fail | **Blank account**  1. Do not input email and password  2. Tap [Submit] button | 1. Failed register notification appeared on the screen on screen , Login screen is not opened |  |  |  |  |
| MC1.3 | **Blank email**  1. Input password, do not input email  2. Tap [Submit] button |  |  |  |  |
| MC1.4 | **Blank password**  1. Input email, do not input password  2. Tap [Submit] button |  |  |  |  |
| MC1.5 | **Not match password**  1. Input correct email form and password and corfirm password didn’t match.  2. Tap [Submit] button |  |  |  |  |

### Log in/ log out function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Sign In** | | | | | | | |
| Step:   * Open Management Console * Login screen will open | | | | | | | |
| MC1.7 | Sign In successfully | **Valid account**  1. Input email and password correctly  2. Tap [Login] button | 2. Home screen opened |  |  |  |  |
| MC1.8 | Sign In fail | **Blank account**  1. Do not input email and password  2. Tap [Login] button | 1. Failed login notification appeared on the screen, home screen is not opene d |  |  |  |  |
| MC1.9 | **Blank email**  1. Input password, do not input email  2. Tap [Login] button |  |  |  |  |
| MC1.10 | **Blank password**  1. Input email, do not input password  2. Tap [Login] button |  |  |  |  |
| MC1.11 | **Invalid email**  1. Input wrong email and password  2. Tap [Login] button |  |  |  |  |
| MC1.12 | **Invalid password**  1. Input correct email and wrong password  2. Tap [Login] button |  |  |  |  |
| MC1.13 | **Invalid authorization**  1. Input email and password of a non-administrator account  2. Tap [Login] button |  |  |  |  |

### User management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **User management function** | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | |
| MC3.14 | Open User management | **Screen open**  1. Tap [User] icon on [Home] screen  2. Tap [back] button on [Home] screen | 1. [User management] screen appeared with [My task],[Contact],[Setting],[Profile],[Log out] buttons.  1. [Login] screen closed and [Home] screen appeared |  |  |  |  |
| MC3.15 | Open User Profile | **Open Profile successfully**  1. Tap [User] icon on [Home] screen  2. Tap [Profile] button on [User management] screen | 1. [User management] screen appeared.  2. Avatar, Fullname. Email information with [Change Full Name] and [Change Avatar] text fields appeared. |  |  |  |  |
| MC3.15 | Changer full name of user | 1. Tap [User] icon on [Home] screen  2. Tap [Profile] button on [User management] screen  3. Tap [Changer Full Name] | 1. [User management] screen appeared.  2. Avatar, Fullname. Email information with [Change Full Name] and [Change Avatar] text fields appeared.  3. You can change Full Name . |  |  |  |  |
| MC3.15 | Change avatar of user | 1. Tap [User] icon on [Home] screen  2. Tap [Profile] button on [User management] screen  3. Tap [Change avatar]  4. Choose a picture and tap [Save] button | 1. [User management] screen appeared.  2. Avatar, Fullname. Email information with [Change Full Name] and [Change Avatar] text fields appeared.  3. You can change avatar . |  |  |  |  |
| MC3.15 | Open User Setting | **Open successfully**  1. Tap [User] icon on [Home] screen  2.Tap [Setting] icon | 1. [User management] screen appeared.  2. [Setting] Screen is appeared |  |  |  |  |
| MC3.15 | Open My Task | **Open successfully**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared |  |  |  |  |
| MC3.15 | Log out User Management | **Log out successfully**  1. Tap [User] icon on [Home] screen  2.Tap [Log out] icon | 1. [User management] screen appeared.  2. [User Management] Screen is disappeared and [Log in] is appeared. |  |  |  |  |

### Project member management function

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | | | **Date** | **Note** | |
| **Project member management function** | | | | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | | | | |
| MC1.8 | Open Project Member screen | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member] | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button |  | |  |  | | |  |
| MC1.8 | Add member | **Add member succesfully**  1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4. Tap [Add Member] button  5. Input [User Name] and tap [Add ] button | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button  4. New member information is dispplayed on Project Member Screen |  | |  |  | | |  |
| **Blank user name**  1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4. Tap [Add Member] button  5. Do not input email . Tap [Add] button | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button  4. Failed add member appeared on screen, home screen is not opened |  | |  |  | | |  |
| **Valid username**  1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4. Tap [Add Member] button  5. Input wrong username . Tap [Add] button | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button  4. Failed add member appeared on screen, home screen is not opened |  | |  |  | | |  |
| MC1.8 | Cancel Add member | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4. Tap [Add Member] button  5. Tap [Cancel] button | 4.Back [Project Member] screen and nothing happen. |  | |  |  | | |  |
| MC1.8 | Remove member | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4. Tap one Member  5.Tap [Remove member from project] | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button  3.Drop list down  4.The Member is removed from Members list of Project. |  | |  |  | | |  |
| MC1.8 | Open Send message box | **Open successfully**  1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4. Tap one Member  5.Tap [Send message to this member] | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button  3.Drop list down  4.[Send message] box is opend. |  | |  |  | | |  |
| MC1.8 | Back to Project Setting screen | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4.Tap [back] button | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button  4.[Project Setting] screen is appeared. |  | |  |  | | |  |

### Project management function

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | | **Result** | **Tester** | | | **Date** | **Note** | |
| **Project management function** | | | | | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | | | | | |
| MC1.8 | Create new project | **Blank Project Name**  1. Tap [Create Project] button on [Home] screen  2. Do not Input Project Name into text box and tap [Save] button | 1. Project Name is not created and nothing happen in here |  | | |  |  | | |  |
| **Create Project Name successfully**  1. Tap [Create Project] button on [Home] screen  2. Input project name on [Home]  3. Tap [Save] button | 1. [Home] screen appeared.  2.[Project] button appeared with correspond name that is created before.  3.My created Project has a symbol, others created Project have not the symbol. |  | | |  |  | | |  |
| MC1.8 | Cancel Create Project | 1. Tap [Create Project] button on [Home] screen  2.Tap [Cancel] button | 1.Back to [Create Project] Screen  2. Project Name is not created and nothing happen in here |  | | |  |  | | |  |
| MC1.8 | Open Project Setting | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting] icon | 1. [Project Setting] screen appeared with [Project Member], [StateBoard Setting] and [Close Project] functions. |  | | |  |  | | |  |
| MC1.8 | Close Project | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting] icon  3. Tap [Close Project] | 1.[Project Setting] screen appeared.  2. [Home ] screen appeared. |  | | |  |  | | |  |

### Stateboard management function

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | | | **Date** | **Note** | |
| **Stateboard management function** | | | | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | | | | |
| MC1.8 | Open StateBoard  Screen | 1. Tap one exist Project button on [Home] screen | 1.[StateBoard] screen appeared. |  | |  |  | | |  |
| MC1.8 | Open StateBoard Setting | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [StateBoard Setting] | 1.[Project Setting] screen appeared.  2. [StateBoard Setting ] screen appeared with [Add Stateboard ] button and list all projects. |  | |  |  | | |  |
| MC1.8 | Rename a State | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [StateBoard Setting]  4.Tap a [Stateboard ] state  5.Drop list down and tap [Rename] | 1.[Project Setting] screen appeared.  2. [StateBoard Setting ] screen appeared with [Add Stateboard ] button and list all projects.  3. [Stateboard] state name is changed. |  | |  |  | | |  |
| MC1.8 | Delete a State | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [StateBoard Setting]  4.Tap a [Stateboard ] state  5.Drop list down and tap [Delete] | 1.[Project Setting] screen appeared.  2. [StateBoard Setting ] screen appeared with [Add Stateboard ] button and list all projects.  3. [Stateboard] state is removed. |  | |  |  | | |  |
| MC1.8 | Add a State | 1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [StateBoard Setting]  4.Tap a [Stateboard ] state  5.Tap [Add Stateboard] button | 1.[Project Setting] screen appeared.  2. [StateBoard Setting ] screen appeared with [Add Stateboard ] button and list all projects.  3. [Stateboard] state is created. |  | |  |  | | |  |
| MC1.8 | Swipe between others States | 1. Tap one exist Project button on [Home] screen  2.Swipe left and right | 2. [StateBoard Setting ] screen appeared .  3.[State] is change |  | |  |  | | |  |
| MC1.8 | Display State | **Display all tasks**  1. Tap one exist Project button on [Home] screen | 1.[StateBoard] screen appeared.  2. All tasks card is displayed |  | |  |  | | |  |
| **Display all members**  1. Tap one exist Project button on [Home] screen | 1.[StateBoard] screen appeared.  2. All members on task is displayed |  | |  |  | | |  |
| **Display Due Date**  1. Tap one exist Project button on [Home] screen | 1.[StateBoard] screen appeared.  2. Due Date is display with one color ( Red – Due time, Yellow – Middle time, Blue- Start time ), |  | |  |  | | |  |
| **Display Comment on Task icon**  1. Tap one exist Project button on [Home] screen | 1.[StateBoard] screen appeared.  2.[ Comment on task ] icon is displayed |  | |  |  | | |  |
|  |  | **Open Comment on Task**  1. Tap one exist Project button on [Home] screen  3. Tap [Comment] icon | 1.[StateBoard] screen appeared.  2.[ Comment on task ] icon is displayed |  | |  |  | | |  |

### Message management function

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | | | **Date** | **Note** | |
| **Message management function** | | | | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | | | | |
| MC1.8 | Send message from sign in account to others account. | **Send message sucessfully**  1. Tap one exist Project button on [Home] screen  2. Tap [Project Setting]  3. Tap [Project Member]  4. Tap one Member  5.Tap [Send message to this member]  6. Input message to [Content ] textbox and tap [Send] button. | 1.[Project Setting] screen appeared.  2. Display all members of the project with [Full Name] and [email] and [Add member] ,[back] button  3.Drop list down  4.Send message box display with receive name of user ,[Content] text box and [Send] button  5.Message is sended to the member |  | |  |  | | |  |
| **Blank message**  6.Do not input message to [Content ] textbox and tap [Send] button. | 5.Message is not sended to the member and nothing happen. |  | |  |  | | |  |
| MC1.8 | Reply message to Sender | **Reply message sucessfully**  1. Tap [Notification] icon on [Home] screen.  2.Tap the message is sended from Sender  3.Input message to [Content ] textbox and tap [Repply] button. | 1.[Notification ] screen display.  2. Message List show message from sender.  3.Reply message box is displayed  4.Message is replied to the sender |  | |  |  | | |  |
| **Blank message**  3.Do not input message to [Content ] textbox and tap [Repply] button. | 4.Message is not replied to the sender and nothing happen. |  | |  |  | | |  |

### Notification management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Message management function** | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | |
| MC1.8 | View Notification screen | 1. Tap [Notification] icon on [Home] screen. | 1.[Notification ] screen display. |  |  |  |  |
| MC1.8 | Messenger Notification Display | 1. Tap [Notification] icon on [Home] screen. | 1.[Notification ] screen display.  2. [Envelope] is displayed, and [Messenger Sender name] is displayed. |  |  |  |  |
| MC1.8 | Open Messenger Notification | 1. Tap [Notification] icon on [Home] screen.  2. Tap one Messenger send from others | 1.[Notification ] screen display.  2. [Envelope] is displayed, and [Messenger Sender name] is displayed.  3. Messenger is opened |  |  |  |  |
| MC1.8 | Remove Member Notification Display | 1. Tap [Notification] icon on [Home] screen. | 1.[Notification ] screen display.  2. [Bell] is displayed, and [Member has remove] is displayed. |  |  |  |  |
| MC1.8 | Add Member Notification Display | 1. Tap [Notification] icon on [Home] screen. | 1.[Notification ] screen display.  2. [Bell] is displayed, and [Member is added] is displayed. |  |  |  |  |

### Task management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Task management function** | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | |
| MC1.8 | My task | **Open Successfully**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared. |  |  |  |  |
| **Display All Project**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared.  3. All project is displayed |  |  |  |  |
| **Display All Member**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared.  3. All members on Task card of each project is displayed |  |  |  |  |
| **Display Due Date**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared.  3. All members on the Project card is displayed |  |  |  |  |
| **Display Edit Task**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared.  3. [Edit task] icon is displayed |  |  |  |  |
| **Open Edit Task**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon  3. Tap [Edit task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared.  3. [Edit task] icon is displayed  4.Edit task is opened |  |  |  |  |
| **Display Comment on Task icon**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared.  3.[ Comment on task ] icon is displayed |  |  |  |  |
| **Open Comment on Task**  1. Tap [User] icon on [Home] screen  2.Tap [My task] icon  3. Tap [Comment] icon | 1. [User management] screen appeared.  2. [My task] Screen is appeared.  3. [Edit task] icon is displayed  4.Comment on task is opened |  |  |  |  |
| MC1.8 | Add task | **Add task screen sucessfully**  1. Tap one exist Project button on [Home] screen  2. Tap [+] button  3.Input [Name] text box.[Start Date] and [Due Date] drop list, assign to some member.  4. Tap [Create] button | 1. [StateBoard] screen is displayed  2. [Add task] screen is displayed  3. Task is created |  |  |  |  |
| **Blank Name**  3.Do not input Task Name to [Name] textbox.  4. Tap [Create] button | 3.Task is not created and nothing happen. |  |  |  |  |
| **Blank Start Date**  3.Do not choose Start Date from [Start Date] drop list.  4. Tap [Create] button | 3.Task is not created and nothing happen. |  |  |  |  |
| **Blank Due Date**  3.Do not choose Start Date from [Due Date] drop list.  4. Tap [Create] button | 3.Task is not created and nothing happen. |  |  |  |  |
| **Blank Assign To**  3.Do not assign any member  4. Tap [Create] button | 3.Task is not created and nothing happen. |  |  |  |  |
| MC1.8 | Description on add Task screen | 1. Tap one exist Project button on [Home] screen  2. Tap [+] button  3. Tap [Add Description] button  4.Input Description . | 1. [StateBoard] screen is displayed  2. [Add task] screen is displayed  3. You can input description sucessfully |  |  |  |  |
| MC1.8 | Edit task | 1. Tap one exist Project button on [Home] screen  2.Tap one [task] card | 1. [StateBoard] screen is displayed  2.[Edit task] screen is displayed  3. Can edit [Name] textbox,[Description] .[Start Date] and [Assignt to] |  |  |  |  |
| MC1.8 | Comment on task | 1. Tap one exist Project button on [Home] screen  2.Tap one [task] card  3. Add comment | 3.Can comment on task with others |  |  |  |  |
| MC1.8 | Assign task | 1. Tap one exist Project button on [Home] screen  2. Tap [+] button  4. Tap one [Assign To]  5.Choose members  6.Tap [back ] button | 1. [StateBoard] screen is displayed  2. [Add task] screen is displayed  3. Show information of Members  4.[Add task] screen is displayed and members are assigned |  |  |  |  |

### Comment management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Comment management function** | | | | | | | |
| Step:   * Open Management Console * Login by valid account | | | | | | | |
| MC1.8 | View comment on task | 1. Tap one exist Project button on [Home] screen  2.Tap one [task] card | 1. [StateBoard] screen is displayed  2.[Edit task] screen is displayed  3.Can view comments on other member task |  |  |  |  |
| MC1.8 | Comment on task | 1. Tap one exist Project button on [Home] screen  2.Tap one [task] card  3. Add comment | 1. [StateBoard] screen is displayed  2.[Edit task] screen is displayed  3.Can comment on task with others |  |  |  |  |

1. **Checklists**
2. **USER’S MANUAL**
3. **Installation Guide**
4. **User’s Guide**
5. **APPENDIX**