**MINISTRY OF EDUCATION AND TRAINING**



MOBILE PROJECT MANAGEMENT

MPM

**CAPSTONE PROJECT**

Specialty: Software Engineering

Project members:

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

Supervisor: Mr. Cao Xuan Vinh

Hanoi – 2013

RECORD OF CHANGES

\*A – Added M – Modified D – Deleted

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

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

Contents

[**A.** **INTRODUCTION** 3](#_Toc357719878)

[**I.** **Project Management System Introduction & History** 3](#_Toc357719879)

[**II.** **Initial Ideal of the Group** 3](#_Toc357719880)

[**III.** **Existing Products& Systems** 4](#_Toc357719881)

[**1.** **2Do: To Do List | Task List** 4](#_Toc357719882)

[**2.** **GTasks: To Do List & Task List** 5](#_Toc357719883)

[**3.** **Air To-Do** 6](#_Toc357719884)

[**4.** **Comparison with our product** 7](#_Toc357719885)

[**IV.** **Our solution and Purposes** 8](#_Toc357719886)

[**1.** **Solution and Improvement** 8](#_Toc357719887)

[**2.** **Our purposes** 9](#_Toc357719888)

[**B.** **PROJECT MANAGEMENT PLAN** 9](#_Toc357719889)

[**C.** **SOFTWARE REQUIREMENT SPECIFICATION** 9](#_Toc357719890)

[**D.** **DESIGN & IMPLEMENTATION** 9](#_Toc357719891)

[**E.** **TEST DOCUMENTATION** 9](#_Toc357719892)

[**F.** **USER’S MANUAL** 9](#_Toc357719893)

[**G.** **APPENDIX** 9](#_Toc357719894)

DEFINITIONS AND ACRONYMS

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| MPM | Mobile Project Management |
| PM | Project Manager |
| PTL | Project Technical Leader |
| TL | Team Leader |
| QA | Quality Assurance Officer |
| DV | Developer |
| URD | User Requirement Document |
| PMP | Project Management Plan |
| SRS | Software Requirement Specification |
| ADD | Architecture Design Document |
| DDD | Detail Design Document |
| SDD | System Design Description |
| TP | Test Plan |
| TC | Test Case |
| UC | Use Case |
| AP | Acceptance Test |
| SC | Source Code |
| IT | Integration Test |
| ST | System Test |
| UT | Unit Test |
| UI | User Interface |
| TR | Test Report |
| GUI | Graphic |
| CM | Configuration Manager |

**Table 0-2: Definitions and Acronyms**

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

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

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

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

1. **Initial Ideal of the Group**

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

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

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

Our team makes final decision that our system will include:

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

1. **Existing Products& Systems**

Currently there are several similar products

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



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

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

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

****

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

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

1. **Air To-Do**

****

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

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

1. **Comparison with our product**

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

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

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

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

**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.

**Cloud Computing**

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

**Benefits of Cloud Computing**

* **Cost Savings**

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

* **Better Scalability and High Performance**

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

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

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

* **Improve Security**

1. **Our purposes**

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

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

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

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

1. **Project Objectives**

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

1. **Project Scope**

The scope of the system:

* Help the project manager in project management by using building project and organization features in our system
* Help the team member in creating and sharing task for others members in the same organization.
* Help the all members can share their comments and experience by using task comments and sending message feature.
* Help a person can find and invite his/her friends for working on a project.
* Help a person in organizing his/her individual work.

Target users of the system:

* Officers working on the project.
* Enterprise companies
* People whose need the task management application

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.
* An account of Microsoft Azure(Educator) for deploying Services and Database
* 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 4.0, 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
* GitHub for Windows Version 1.0.48.0

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

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

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

1. **Techniques**

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

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

1. **Project Organization**

### Software Process Model



The process model used for developing this project is iterative and incremental development model. We have only approximately 3 months for doing everything of this project and not every member has experience for that similar system before. Beside, the beginning the problem definition is not totally clear and we need to clarify it in iterative model. The iterative way lead us to reduce the misunderstand requirements and make sure project runs in the right way. So we should choose this iteration rather than water fall model. Furthermore, incremental model can help us to break down many phases into small steps and do some steps in parallel. Working in parallel some steps can help us to reduce effort and project can delivered on only 3 months

### Roles and responsibilities

1. **Product Life Cycle**

### Project stage summary

### Project stage and milestone

1. **Project Management Plan**

### Tasks

### Task Sheet: Assignment and Timetable

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