**UL HOUSING PROJECT: Project Management Plan**

**CMPS 453**

**Dr. Kumar**

**Fall 2014**

Issa Samake

Zach Danjean

Jaquincy Nelson

Yee Wong

Brandin Jefferson

Brian Okoye

**Table of Contents:**

* **Introduction 3**
* **Project Organization 4**
* **Life Cycle Model 4**
* **Risk Analysis 5**
* **Hardware and Software Resource Requirements 5**
* **Deliverables,Schedules 6**
* **Monitoring, Reporting , Controlling Mechanisms 7**
* **Professional Standards 8**
* **Evidence of Configuration Management 8**

**INTRODUCTION**

Our group was given the task of creating a way to organize U.L housings individual residents data. We were asked to to create an electronic way of keeping track of U.L. Housing residents request and Housing Application information. As well as keep track of room residency within each room U.L housing is responsible for on campus. This is to create a more seamless process in how housing will deal with pulling up information on residency and request. This is to free up time for the employees of U.L housing so they can be able to attend to other important task, as well as free up space within the housing building occupied by physical storage containers. Ease of accessibility of various information pertaining to U.L residence will lower the latency in which processes are handled.

Our goal is to turn housings already existing method of dealing with applications, room changes, room maintenance, and room occupancy information, into one application that deals with all these task. The application will be designed to be user friendly for the residence as well as housing employees. For residence we will create a database that will hold all of each individual residence information pertaining to the above task specified earlier. The information will be organized by a library of residence clid numbers. A resident with a specific clid number can request maintenance room changes, or an aspiring resident can fill out a housing application to be joined into the system. This will be handled with various interfaces, pertaining to either one of the mentioned task. Once a user fills out the needed information into the interface, the then filled out information will be stored under the users clid number in the library. From the information stored in this library other databases will be affected. For example if a resident fills out the application to request a specific room and the application is approved, the database holding the rooms and occupancy of the specific rooms will increment a count specifying that the rooms occupancy state has been changed. Room occupancy will change based on the approval of new applications or room changes. If a resident lease is voided or leaves for any specific reason the application room counter will be designed to deal with that as well. Each individual room will also be held in a library based on building and room number This is were the count will be stored, and if the count specifies a specific number value the rooms number will then be placed within another table of available rooms. Request such as maintenance will also be placed in a table with clid user and room number tagged to each specific row as well. This information can then be used for the housing employee's to determine which rooms are available to be occupied for existing residents seeking a room change or for new residents being placed in rooms, also what rooms are awaiting maintenance based on a method connected to the room and resident database.

The structure of how the application will be developed begins with organizing the project, and then discuss the life cycle model used. We will then analyze the potential risk within development, and also discuss the hardware and software that will be used for application. We will then establish a schedule for the activities. Then we will explain the different monitoring, reporting and controlling mechanisms used. And finally we will explain what is expected out of each individual member of the development team.

**PROJECT ORGANIZATION**

The team has six members: Issa Samake, Zach Danjeau, Jaquincy Nelson, Yee Wong, Brandin Jefferson, and Brian Okoye. Documentation will be handled by all team members by splitting the document evenly; the same approach has been decided for use with coding as well.

Responsibilities

Brandin Jefferson:

* Primary: User Interfaces
* Secondary: Database (File Storage), Program Documentation

Issa Samake:

* Database (Conversion), General Documentation

Zach Danjeau:

* Primary: Hardware (File scanning and conversion)
* Secondary: Database

Jaquincy Nelson

* Primary: Map construction (Map interface)
* Secondary: General Documentation

Yee Wong

* Database (Formatting/Skeleton), Program Documentation

Brian Okoye

* Team Contact, Database (Formatting/Skeleton)

These roles were chosen with each individual’s current skills in mind. If a team member had experience in a certain area, he was given a task that fit said experience. This lessens the work load by having each member only need to learn a new programming language, rather than several concepts at once. This will also allow more time for development outside of the planning stages.

**LIFECYCLE MODEL USED**

The software development lifecycle model used for this project is a mix of the waterfall and agile software development model. We used the waterfall model at the early stage of the project for the planning as well as for writing the requirements. We chose to use the waterfall model at the beginning because it was easy to understand and follow and it produced better requirements.

We used the agile model for the design, development and testing stage of the project. We used it at those stages because it was more flexible and allowed stages to overlap and run concurrently making the whole process faster. Also, it allowed adaptation to changes of requirements from the customer.

**RISK ANALYSIS**

The risk for the project include the possibilities of group members losing motivation, group members dropping out of the class, and group members deciding that they want to switch teams. This would result in staff turnover and management change risks within the project. The best way to handle these risks is for the every member in the group to try their best to complete the project. Motivation will be a combination of a finished project and a good grade. This should also keep members from wanting to switch teams.

**HARDWARE AND SOFTWARE RESOURCE REQUIREMENTS**

The hardware that will be used to carry out the development will be the computers in the CMPS lab, running on windows 7 virtual machine. Development will be also carried out on each group members individual personal laptops. Printers in the Cmps lab and other labs around U.L Lafayette campus will be used to print out documentation. We will use programming software as well as database software to implement code for the development. The Ide's will be used for interfacing and will be used alongside html script and databasing software. Ruby on the rail will be the programming language of choice as well as Mysql to help implement a database. Git will be used as a repository for the applications development. Web applications such as google doc. , and google hangout will be used as a means of communication for members within the group. Ul Lafayette email will be used to communicate with client when necessary. Facebook messenger, application will also be used for communication.

**DELIVERABLES, SCHEDULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Task Name** | **Date of competion** | **Ressources (Human)** | **Dependencies** |
| 1 | Define the problem | 9/19/14 | 6 |  |
| 2 | Analyze the problem | 9/25/14 | 6 | 1 |
| 3 | write the requirements documentation | 9/30/14 | 4 | 2 |
| 4 | Determine Technologies and tools to be used | 9/30/14 | 6 | 1 |
| 5 | Start training with new technologies and tools | 10/8/14 | 6 | 4 |
| 6 | Write project plan | 9/30/14 | 2 | 2 |
| 7 | Design solution | 10/9/14 | 4 | 3 |
| 8 | Write architecture Documentation | 10/9/14 | 2 | 7 |
| 9 | Development phase | 11/10/14 | 5 | 8 |
| 10 | Write Test Plan | 11/15/14 | 2 | 9 |
| 11 | Tesing and Debugging phase | 11/25/14 | 2 | 10 |
| 12 | Deployment | 11/26/14 | 6 | 11 |
| 13 | Final Project Report | 11/27/14 | 6 | 10 |

**MONITORING, REPORTING, AND CONTROLLING MECHANISMS**

The controlling mechanisms should compare progress with targets and standards. When satisfactory, the process with keep running. But when unsatisfactory, some corrective actions will be taken.

The first step in controlling the projects will be to develop standards of performance and quality, measurable goals and milestones. The next step is to monitor the project process based on the previous criteria. The project is under control when the team achieve the goals and arrive to the milestones on time and with the quality expected . If not, the project is out of control and needs some actions to be taken. If the delay is not significant, a correction can be made before moving forward in the process. However, if the delay is significant, the team will need to revise the plan.

In order for the monitoring systems to be effective, we will need a reporting mechanisms.

The reporting mechanisms will be composed of two types. The internal reports which will be mostly done by weekly team meeting when each team member will inform the other team members of his progress on the project .The external reports will be done to inform the instructor (manager) and the clients about the team progress on the projects. The team will use log sheets, progress reports as well as deliverables such documentation as external reports.

**PROFESSIONAL STANDARDS**

**Scholastics Dishonestly :**

Team members programming material should be completely of their own work . On the first occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.On the second incident , the team will notify the instructor of the problem and a meeting will be setup to resolve the current issue. On the third occurrence, the instructor will be notified and a meeting will be held ; The team may have an option of removing the offending member.

**Meeting Schedules :**

Each team member should attend a majority of team meetings. Incase of a valid reason, a team member/s may setup another meeting to review any prior meeting scheduled.On the second incident , the team will notify the instructor of the problem and a meeting will be setup to resolve the current issue. On the third occurrence, the instructor will be notified and a meeting will be held ; The team may have an option of removing the offending member.

**Quality Expectations :**

Each member should attempt to execute a higher quality of work based on the task at hand .

On the first occurrence of a member lacking in work quality, a team meeting will be held and resolved by team discussion . On the second incident , the team will notify the instructor of the problem and a meeting will be setup to resolve the current issue. On the third occurrence, the instructor will be notified and a meeting will be held ; The team may have an option of removing the offending member.

**EVIDENCE OF CONFIGURATION MANAGEMENT**

Git Repository: https://github.com/brandinjefferson/CMPS453-Docs