**JAIL INFORMATION MANAGEMENT SYSTEM**

Review Milestone

Requirements Analysis and Project Plan Report Content

September 30, 2015

Team Members

Print Name: Joseph Comeau

Signature:

Print Name: David Edwards

Signature:

Print Name: Mike Pasek

Signature:

Print Name: Eric McCoy

Signature:

Print Name: Erik Micklesen

Signature:

Project Plan Contents

**Project Team**

David Edwards – Testing

Joseph Comeau – Team Manager

Mike Pasek – User Interface

Eric McCoy – Development Activities

Erik Micklesen – Development Activities

**Development Environment**

Platform/Operating System:

Development:

OS independent web framework: Unix/Mac OS X/Windows

Localhost web server

Production

Hosted Linux machine accessed via the Internet

Implementation Language:

Python

JavaScript

HTML/CSS

Database Application

Django object relational mapping (ORM)

SQLite for testing

PostgreSQL or MySQL for production

Rationale for Selection

The Django web framework was chosen because it handles most of the backend issues for deploying applications to the web. The framework has template language for designing HTML documents that speeds up the design process. Built-in security features.

**ICSM Common Case to be used for development and rationale:**

Software-intensive application or system – *software application/system that executes on one or more commercial hardware platforms. It can be a stand-alone software system or a constituent within one or more systems of systems*.

A software intensive application/system is a system whose key contributors are software features which influences the overall design, feel, deployment and evolution of that system. We decided to choose software-intensive application/system as our common case because it is more suited for the task at hand. The Jail Management System will not require any new advancement in hardware and can be run on a standard computer that has access to the Internet. Users will not be required to update their systems hardware in order to use the product, which will greatly decrease the overall cost of the system. Further, by designing an application that can be accessed via the Internet it allows users to use the product remotely, which can be beneficial to police officers, doctors, etc. who are not always at the jail. Maintenance (software updates) can be more easily applied when there is a limiting amount of hardware dependencies.

**Methodologies/techniques to be used in development and rationale:**

Life Cycle Model: Incremental Life Cycle Model



The incremental life cycle model consists of dividing up requirements into various builds, where each subsequent release adds an additional feature to the overall system. Our rationale for choosing the incremental model stems from the fact that there is limited time to develop the product (3 months) and we already have a clearly defined list of requirements. With a good understanding of the major requirements it allows us to begin building a rough version of the product and we can change/add features as more information is gathered from the customer at every new build.

Advantages:

Less costly to change requirements because of the smaller iterations

Easier to test and debug

Customers can weigh in on each build

Easier to identify/handle risks because of the smaller iterations

Process Methodology: Extreme Programming (XP)

XP methodology is suited for small teams that work in close proximity of one another to allow frequent communication between team members, which works well for us because we all live in the same area. By having small iterations, we will be able to get rapid feedback from unit tests and customers. XP will help us keep the product simple and clean, which will be beneficial due to our time constraint.

Basis of Development Estimate – Software Size:

We are using source lines of code (SLOC) to estimate the size of our software. Note: the following calculations are from COCOMO II…

New lines of code = 250

Reused lines of code = 10,000

Modified lines of code = 0

Total Equivalent Size = 250 SLOC

Component breakdown:

|  |  |  |
| --- | --- | --- |
| **Component** | **# of lines** | **Complexity** |
| Booking | 125 | Low |
| Release Process | 25 | Medium |
| Alert system prior to the release date of an inmate | 50 | High |
| Housing of inmates | 25 | Medium |
| Sentence Calculation | 25 | High |

Basis of Development Estimate – Software Size:

The total number of labor hours required for the first increment will be around 3.2 months. This estimate is based on the software size and the schedule we have come up with above. Requirements may have to be adjusted due to the fact that we are all part-time.

**Requirements**

Statement of Scope of Effort/Understanding of Problem:

For the Jail Information Management System, we will be addressing several key functionality requirements. The overall system we are creating will provide: a full graphical user interface, the capability to support multiple users in a web environment, a password secure system, login information to be entered only once, validated information entry, name searches (with wildcard characters), online help capability, a menu-driven user interface, and error messages when given improper information.

As part of the processing of inmate information, a user must have the ability to input general information regarding the respective inmate. This includes fields such as name, aliases, height, weight, social security number, etc. Users will also be able to input and access any medical information for each inmate such as any health issues, medications, and other health related issues. When an inmate is booked, the system will calculate the sentence duration and shall process all information regarding their release date. Additionally, the system will store information regarding each inmate’s personal property, as well as their financial accounts while booked.

The complete functionality of JIMS is limited to certain users. This is crucial in maintaining proper security throughout JIMS. For example, the jail doctor is denied access to an inmate’s property and financial information. Likewise, police officers and sheriffs are limited to the general and medical information regarding an individual, where as the jail system administrator is allowed access to all areas in the system (ie. inmate property management, inmate financial accounts, release processing, etc.).

Per system performance requirements, JIMS shall respond within 5 seconds of any command. When a user inputs information for a page, confirmation or denial of valid input should be instantaneous, but never exceeding a 5 second response time. This 5 second requirement also includes navigating the system from page to page.

For the first increment, we will be addressing the high priority requirements that are essential to getting the system up and running. This includes providing a basic graphical user interface and generating inmate release information. Also, information such as gang affiliation and inmate background history must be stored to determine the compatibility among inmates. For example, inmates in rival gangs cannot be housed in close proximity. The first increment will clearly not have complete functionality, but it will provide users enough capability to process and book inmates.

Discussion of completeness, consistency, testability, etc. of requirements:

All of our source code will be tested using unit testing to ensure that the basic functionality works appropriately. Along with unit testing, we will take advantage of Selenium WebDriver to automate the testing of our user interface. Also, we are developing our test cases in the very beginning of the increment, so that we can map out all the requirements and necessary functionality that is needed to satisfy the customer/user. By using test cases as a road map for our development, it can help prevent any deviations in the product, not allowing us to become sidetracked on a requirement/feature.

Total Number from initial requirements document by functional area:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Functional Area** | **Total # Requirements** | **High Priority** | **Medium Priority** | **Low Priority** |
| System and Performance Requirements | **5** | 3 | 1 | 1 |
| Framework | **1** | 0 | 1 | 0 |
| Detention Processing Requirements | **67** | 43 | 23 | 1 |
| Trust Accounting Requirements | **0** | 0 | 0 | 0 |
| Inmate Property Requirements | **0** | 0 | 0 | 0 |
| Inmate Release | **5** | 0 | 5 | 0 |
| Diposition/Sentence Calculation Requirements | **0** | 0 | 0 | 0 |

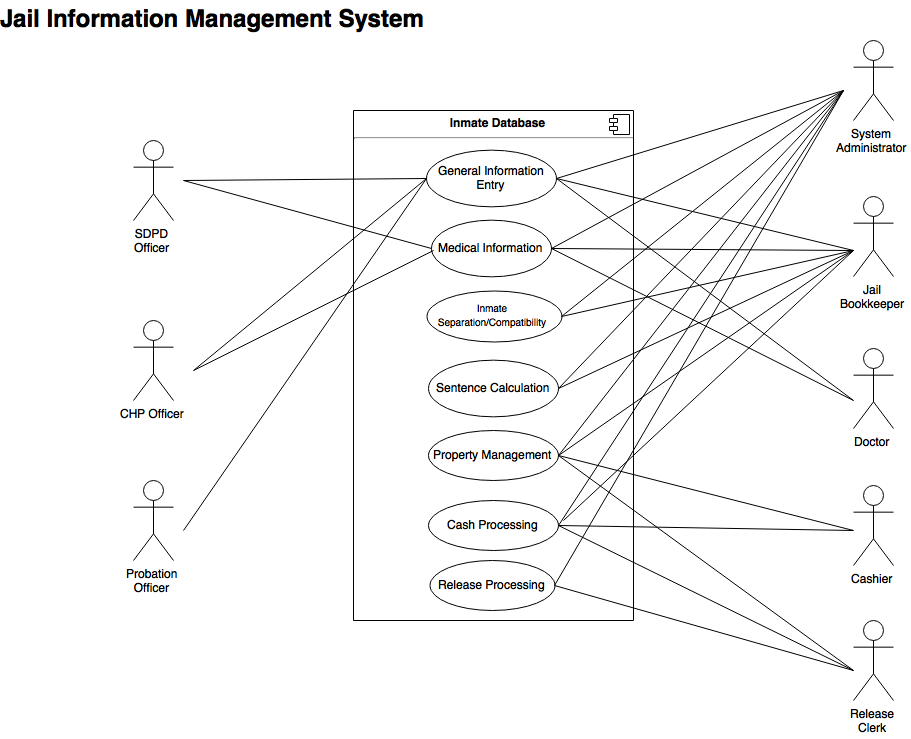
Number of requirements added, deleted, or changed during requirements analysis activities:

We have added one additional requirement to our first increment. That requirement was related to the housing of the inmate (cell number/cell block the inmate lives in). We also changed one requirement temporarily for this increment, which was the sentence calculation. For this requirement we only needed to allow a comment area, where the user can manually enter in the required sentence. Lastly, we did not delete any requirements, but since we are limited in time and team members, we may need to renegotiate with the customer some of the lower priority features.

Total number of requirements going forward into top-level design:

Out of the 78 requirements for the first increment, we are expected to reach 90% completion, which results in 70 of 78 requirements being met. If all the requirements are not met for this increment, the ones missed will be of lower priority.

**Unified Modeling Language Diagram**



**Risks**

One of the biggest challenges to the project is the limited scheduling of each of the team members. With full school schedules and work, the members of the JIMS team must be able to manage and complete the tasks for each increment. Seeing that each subsequent increment is dependent on completion of the previous increment, it is crucial that the team completes each of the increments with a nearly 95% completion rate.

In addition to a limited schedule, most of the team members are new to the Django framework. It will take some time for these members to familiarize themselves with it. Because of this new methodology, it may pose a minor risk to the development of the project.

One area which will require further analysis is the sentence calculation. This was initially forgotten by the customer, and was added to the requirements after the first meeting with the customer. The sentence calculation is also highly complex, with many factors needed to formulate the release date. Although this is high priority on the list of requirements, it will take considerable time to fully understand and develop to ensure proper calculation of sentencing.

In order to implement the sentence calculation feature, the JIMS team must meet with the customer to ensure complete understanding of the formula. After clarification, the team can proceed to develop and integrate this capability into the system.

With a smaller team size and limited time, the JIMS team must organize and use their time wisely in order to hit deadlines and complete milestones.