**Senior Project Website**

**Version 5**

**Feasibility Study and Project Plan**

CIS 4911 Senior Project

Section U01

Professor Masoud Sadjadi

**Prepared by**

Jacek Kopczynski

Yamel Peraza

April 20th, 2015

# 

Copyright

*All title, including but not limited to copyrights, in and to the SENIOR PROJECT WEBSITE are owned by School of Computing and Information Sciences Faculty of Florida International University. All title and intellectual property rights in and to the content which may be accessed through use of the SENIOR PROJECT WEBSITE is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by School of Computing and Information Sciences Faculty of Florida International University.*



Abstract

*The purpose of this document is to conduct a feasibility study of the current "Senior Project Website". In this study, the deficiencies and limitations of the system will be identified and solutions will be proposed as part of the new system. These solutions will not only address the opportunities present in the current system, but will also enhance the system by proposing additional components that are significant enough to provide a better user experience for mentors and students. Furthermore, this document contains a description of the team’s general plan.*

**Table of Contents**

*Senior Project Website V5*



[Copyright 2](#_Toc393537026)

[Abstract 3](#_Toc393537027)

[1. Introduction 6](#_Toc393537028)

[1.1. Problem Definition 6](#_Toc393537029)

[1.2 Background 8](#_Toc393537030)

[1.3. Definitions, Acronyms, and Abbreviations 8](#_Toc393537031)

[1.4. Overview of Document 10](#_Toc393537041)

[2. Feasibility Study 11](#_Toc393537043)

[2.1. Description of Current System 11](#_Toc393537044)

[2.2. Purpose of New System 12](#_Toc393537046)

[2.3. High-level Definition of User Requirements 13](#_Toc393537047)

[2.4. Alternative Solutions 15](#_Toc393537048)

[2.4.1. Description of Alternatives 15](#_Toc393537049)

[2.4.2. Selection Criteria 17](#_Toc393537050)

[2.4.3. Analysis of Alternatives 18](#_Toc393537051)

[2.5. Recommendations 19](#_Toc393537052)

[3. Project Plan 21](#_Toc393537053)

[3.1. Project Organization 21](#_Toc393537054)

[3.1.1. Project Personnel Organization 21](#_Toc393537055)

[3.1.1. Hardware and Software Resources 23](#_Toc393537056)

[3.2. Identification of Tasks, Milestones and Deliverables 24](#_Toc393537057)

[4. Appendix 25](#_Toc393537058)

[4.1. Appendix A – Project Schedule 25](#_Toc393537059)

[4.2. Appendix B – Feasibility Matrix 26](#_Toc393537061)

[4.3. Appendix C – Cost Matrix 27](#_Toc393537062)

[4.1. Appendix D – Diary of Meetings 28](#_Toc393537063)

[5. References 29](#_Toc393537064)

1. Introduction

*Senior Project Website V5*



This section introduces the “Senior Project Website: Version 5” pertinent document information. The first part of the document defines the problem addressed on this current version. In addition, background on previous versions will be discussed and an overview of the entire document will be included for ease of navigability.

The second part of the document will cover the feasibility study, including description and purpose of the current system and a high level definition of the user requirements. The third and final part of the document will discuss the organization of the project, hardware and software used resources.

* 1. Problem Definition

For Senior Project class, students are required to use virtual machines to develop and deploy their applications. The process of requesting virtual environments currently is done manually, given that Senior Project Website does not have the functionality of processing or managing virtual machine requests for its users. The impractical approach of manually finding each student’s email requesting virtual machines is time-consuming, lacks efficiency and is error prone.

The head professor would like to extend the usability of the system by allowing students to request and customize their virtual machine settings, while letting him manage the students’ requests.

We also plan to extend the functionality of the site by adding smaller features like: forgot password, adding students with incomplete grades, changing passwords through act as user, and fixing problems with functionality of the old site.

* 1. Background

The current version of Senior Project Website does not offer the functionality of requesting virtual machines to the head professor. As a consequence, a brief description of how this task is currently performed is discussed in this section.

Students enrolled in Senior Project class are automatically signed up for the Senior Project Website and algorithmically assigned into a team-project. After the projects have been assigned to every student, the head professor sends an email to all active students in the class requesting their needs of virtual machines environments for the completion of their project. Then, each student that needs virtual environments replies to the head professor’s email with all their needs regarding virtual machine. Finally, the head professor tracks each individual’s email, gathers all the information, and sends it to the school system admin, who will create the virtual machines for the students.

There is a lot of manual work to be done to collect and send the request of virtual machines for each individual team. Students in previous semesters of Senior Project class have not been able to develop a functionality into Senior Project Website that addresses this problem. In this project, we will try to resolve this issue.

* 1. Definitions, Acronyms, and Abbreviations

EULA: End-User License Agreement

FIU: Florida International University

Google Docs: Free web-based office suite offered by Google within Google Drive service.

PHP: Open source server-side scripting language designed for web development to produce dynamic web pages.

SPW: Senior Project Website

SPWv.1: Senior Project Website Version 1

SPWv.2: Senior Project Website Version 2

SPWv.3: Senior Project Website Version 3

SPWv.4: Senior Project Website Version 4

**SPWv**.5: Senior Project Website Version 5

**NRMP**: A form of matchmaking devised by the national residency matchmaking program (NRMP). The details of it are that hospitals have residency spots to fill and a ranked list of applicants they want filling them. Applicant ranked residency programs themselves. The matchmaking process works by having applicants apply to residencies in their list’s order. If another applicant tries to displace a current tentative applicant they displace the tentative applicant if the challenging applicant is more desired by the program else they try the next program, displaced applicants try their next program too. Matching ends when all applicants are match, or all applicants are match minus ones who went through their entire list.

**Heuristics**: Colloquially means “common sense approach”, in computer science the meaning is adapted to mean a technique applied to solve a problem. I.e. to shorten average job time in a computer do short jobs first.

**VIP**: Very important project, a project ranked by the head professor to be of a score between 2 and 100 this project will undergo intense matchmaking using heuristics to find a perfect team quickly (higher scored means higher priority)

**OP**: Other project, a project ranked by the head professor to have a score of 1, such projects will undergo NRMP matchmaking to give the ability to compromised project proposer and student constraints. These are more hands-off.

**Agile methodologies:** A method of software development where features are done on an iterative basis with increments on a time period by time period basis. I.E. a developer aims to do feature X this week they do it and they might do more, or they fail to do it and will require more time. Obviously stalling too much on a feature might lead to consequences.

* 1. Overview of Document

Section 1 Introduces the SPWv.5 project goals.

**Section 2** Discusses the feasibility study, meaning how feasible the proposed system is versus already developed technology and other alternatives.

**Section 3** Covers the project plan and organization

**Section 4** is an appendix of supplementary information for prior sections.

**Section 5** covers the references in this document.

1. Feasibility Study

*Senior Project Website V5*



The feasibility study analyzes alternative solutions to the various problems. For each alternative solution, benefits and limitations, will be examined. After a detailed analysis, we will be able to determine the best possible solution. We will also describe the current system and the purpose of the proposed solution. Finally, the study will conclude with justifiable recommendations.

* 1. Description of Current System

The current system, Senior Project Website Version 4, does not provide any means of requesting virtual machine environment to the head professor. The above section (1.2) explains how this task is fulfilled.

What is wrong with this approach?

While this approach does fulfill the necessity of virtual machines creation for each individual student, it lacks of an efficient way of requesting them. Not to mention that there is not a standard set of parameters or settings for the request of the virtual machine environments. In addition, there is not a stable way of keeping track of students and virtual machine association. As a result, there is a lot of manual work that has to be done to collect the virtual machines needs for each individual team.

* 1. Purpose of New System

Senior Project Website Version 5 aims to implement a method of requesting virtual machines to the head professor, providing the users of the application with a more efficient way of satisfying their needs. The implementation of this new feature and the eradication of the existing system’s bugs will ensure that both, students and professors, have a better and more efficient web-tool for Senior Project class.

Overall, the new system will allow the head professor and students to create and track their virtual environments requests without the need of third parties application interaction. Also, it will provide the head professor with a management tool for virtual machine requests.

* 1. High-level Definition of User Requirements

**Version 1:**

* Project members can edit their project.
* Project members will be able to invite students to their team.
* Project members will be able to invite other users as mentors to their team.
* Project proposals will include the technologies to use, in a tags-like way.
* The Head Professor will be able to approve or reject projects.
* Users will be able to register using their LinkedIn accounts.
* User will be able to pull their technology skills from LinkedIn
* The system will automatically present users with a list of suggested projects
* The system will be able to present users a list of suggested users (as mentors or students) for their projects
* Users will be able to search for users, skills and projects.

**Version 2:**

* Head Professor will be able to set the internal system deadline.
* An API will be implemented to allow access to the site only to students registered on Senior Project Class , create/update/remove d students profiles and provide info about students and their projects titles to the Mobile Judge and Online Job Fair projects.
* Guest access will be provided for those students who are not registered on the class.
* Registered users will be able to upload profile pictures from local storage.
* Student will able to upload, download files to the set of shared project files.
* Mentors will be able to upload, delete or download any file that has been uploaded to the server.

**Version 3:**

* Best Fit Project Algorithm: calculates the best possible student-project match.
* Project Repository, where the Head Professor has full access to all projects.
* Users may share documents with project scope.
* Users may download a zip folder of the entire project.
* Head Professor may use the Files Repository without the use of milestones.
* Manage Users: create new mentors manually, or send an invitation email with a unique link for verification. Edit or delete existing users in any role, especially inactive students and mentors from previous semesters.
* Head Professor can override users on SPW.
* Allow the Head Professor to impersonate any user and act on his behalf.
* Aesthetics and Efficiency Fixes:
* Add side scroll arrows to view full description of all projects sequentially.
* Modify login window format.
* Give SPW a responsive design for optimal viewing across a wide range of devices from mobile phones to computer monitors.

**Version 4:**

* Partition the defined constraints of matchmaking
* Increase matchmaking speed
* Give better feedback to matchmaking’s effect to head professor
* Give some more customization options in matchmaking for the head professor

**Version 5:**

* Students are allowed to create several virtual machine requests
* Head professor can modify the settings of the virtual machine requested
* Head professor is able to delete a virtual machine request
* Head professor is able to accept and/or deny student’s virtual machine requests
* Head professor is able to filter virtual machine request
* Head professor is able to notify the school’s system admin via email from the website
* Head professor is able to add new image’s names to the system
* Head professor is able to edit an existing image name in the system
* Head professor is able to change the status of an existing image in the system
* Head professor is able to delete an existing image name in the system
  1. Alternative Solutions

This section will describe alternatives solutions to the problem. A selection criterion will be analyzed based on the feasibility of such alternatives. Finally all the alternatives will be compared concluding on the best option to pursue.

* + 1. Description of Alternatives

Alternative I: Current System

Keep the current practice of requesting the virtual machine environments manually. Although it is not an efficient way of requesting the student’s needs, it is proven to work. The main downside of this practice is that there is not a stable way of tracking student and virtual machine association. For example, emails requests can be lost or forgotten, implying the student has to submit a new request via email, etc.

Alternative II: Proposed System

Implementation of the proposed functionality for Senior Project Version 5. This proposed solution will efficiently allow users of Senior Project Website to request the creation of virtual machine for the completion of their project and will allow the head professor to have a much more organized web-tool to manage Senior Project class.

* + 1. Selection Criteria

The previous alternatives will be analyzed based on their operational, technical, schedule, and economic feasibility. Following criteria utilized from SPWv.4 Feasibility study.

Operational Feasibility

The operational feasibility deals with how functional an alternative is in dealing with the system requirements and how well it is received. An ideal solution would be user-friendly and well received. Is it easy to use? Does it solve the problem? Does it provide the users with accurate and desired information? Does it generate positive feedback by its users?

Technical Feasibility

The technical feasibility focuses on understanding the available technical resources and how they can be used to implement the system. Is the builder team familiar with the technologies? How well do they need to master the technologies? Are the technologies supported at the moment or/and in the future?

Schedule Feasibility

The schedule feasibility deals with the amount of time needed to implement an alternative. An ideal alternative will be quick to produce. How long will this solution take to design and implement?

Economic Feasibility

The economic feasibility analyzes the cost and benefits of an alternative. An ideal alternative will be inexpensive to produce. How much will it cost to implement? Will there be any gains? Will it have additional costs in the long run?

* + 1. Analysis of Alternatives

Analysis of the metrics outlined in section 2.4.2 are as follows:

Alternative I, the current system is rather inefficient, making it hard to track all the students’ emails that are requesting virtual machines. Also it is time consuming when the head professor has to group all the virtual machine request made by students and send them via email to the school’s system admin.

Alternative II, the proposed system will potentially increase the functionality of the system and its efficiency without incurring in any economic expense. This proposed updated system will increase functionality and therefore be convenient for both students and professor. The cost effectiveness of this solution allows it to be done in a period of a semester to ensure timely accessibility by students.

* 1. Recommendations

The SPWv.5 recommends Alternative II as the proposed system because this system will guarantee that head professor and students can manage their virtual machine requests in an efficient way. Additionally, Alternative II does not incur in any economic expense and is feasible for students to complete in the span of a semester. Alternatively, Alternative I (even though it works) is mainly manually done and in some instances it can introduce problems regarding the management of the virtual requests. As a result, Alternative II is a more efficient choice.

1. Project Plan

*Senior Project Website V5*



This chapter will cover the project plan for the SPWv.5, personnel organization, hardware and software requirements, milestones of the project and the costs involved for its development.

* 1. Project Organization

This section details how the SPWv.5 will be organized throughout the life of the project. It also contains the technical needs of the project at this time; this includes the hardware and software needs of the developers.

* + 1. Project Personnel Organization

Even though two members of the team will work under the same project name, this project will be a one person-project because each individual will have different responsibilities and tasks at hand. The duration of the developmental part of the project is spread over 14 weeks and the agile management methodology tool used is Mingle.

A two weeks sprint length is planned for a total of five sprints. Every sprint except the last one will focus in developing specific user stories. Daily meetings will be conducted with the team’s members to talk about what was done the previous day, what needs to be done in the present day and any challenges we are facing to complete the tasks for the current sprint. Also, meetings with the product owner at the beginning and end of each sprint will be scheduled to discuss the progress of the project. After sprint number 5 the focus will be placed on the deliverables seen under tentative schedule in CIS moodle.

|  |  |
| --- | --- |
| Name | Role |
| Jacek Kopczynski  Yamel Peraza | Responsible for ensuring the project progresses based on specifications from client and advice from mentor. Leads discussion ideas and adapts to client needs. |

**Table 3.1** – Roles

|  |  |
| --- | --- |
|  | Jacek Kopczynski Yamel Peraza |
| Project Plan | ✓ |
| Deliverables | ✓ |
| User Interface/Views | ✓ |
| Database | ✓ |
| Controllers/Models | ✓ |
| User Authentication |  |
| Testing | ✓ |
| Final Presentation | ✓ |
| Demo | ✓ |

**Table 3.2** – Project preparation and development roles (Note: “✓” denotes participation in role)

* + 1. Hardware and Software Resources

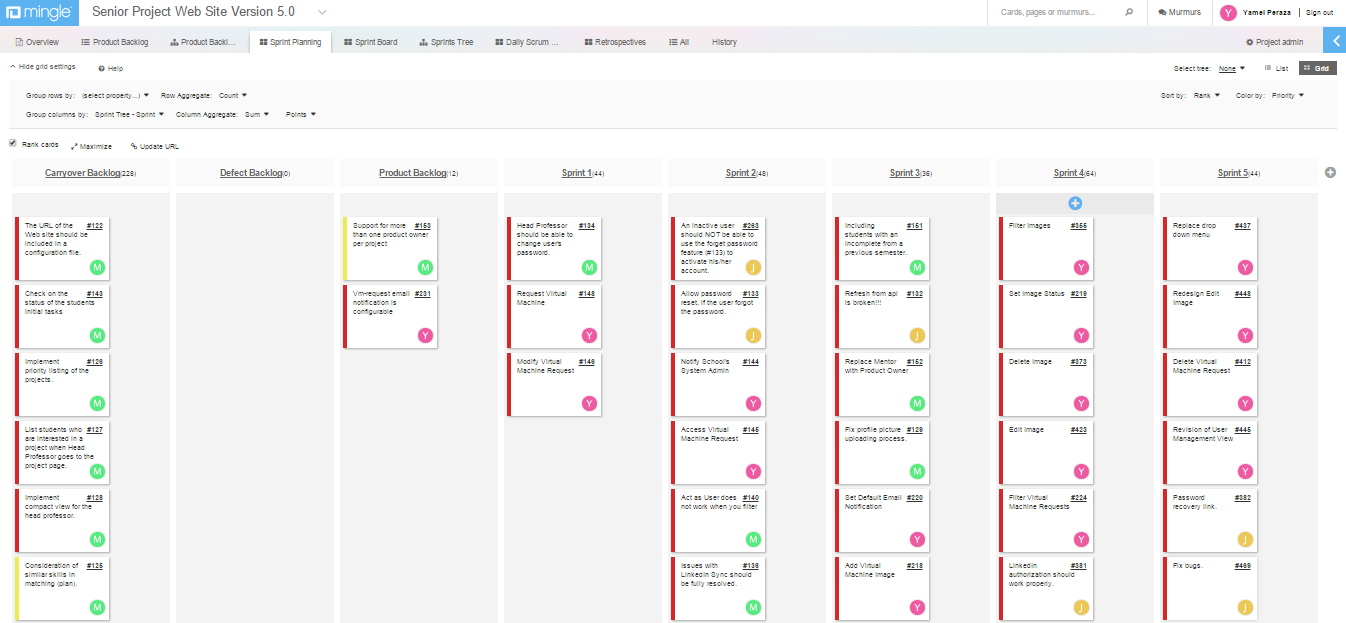
This section lists the software and hardware needed to develop and deploy the proposed project. Hardware and software will not change from version 1 to version 5.

|  |  |
| --- | --- |
| Software | Hardware |
| Sublime Text 2 | Processor: x86 / x64  2.0 GHz or faster |
| PHP 5.1 | Memory:  2 GB DDR3 or higher |
| MySQL 5.0 | Disk Space:  20 GB |
| Microsoft  Office Package 2010 | Display:  1024 X 768 with DirectX 9 |
| Mac OS X |  |
| Microsoft Windows 7  Home/Pro/Ultimate |  |
| Apache 2 |  |
| Firefox Mozilla  ver. 16.0 or higher |  |
| Internet Explorer  ver. 7.0 or higher |  |
| Safari  ver. 5.0 or higher |  |
| Google Chrome  ver. 19.0 or higher |  |
| Skype / Team Viewer |  |
| GitHub repository |  |

Note: This table is from the SPWv.1 “Feasibility Study & Project Plan” (Fernandez, Sanchez, Moya 1).

* 1. Identification of Tasks, Milestones and Deliverables

The tasks and milestones for this semester have essentially occurred on a weekly basis. The end result being the virtual machine system outlined in this proposal. All deliverables are to be turned in at the end of the semester. Also, deliverable-relevant content is to be created during the different sprint periods.

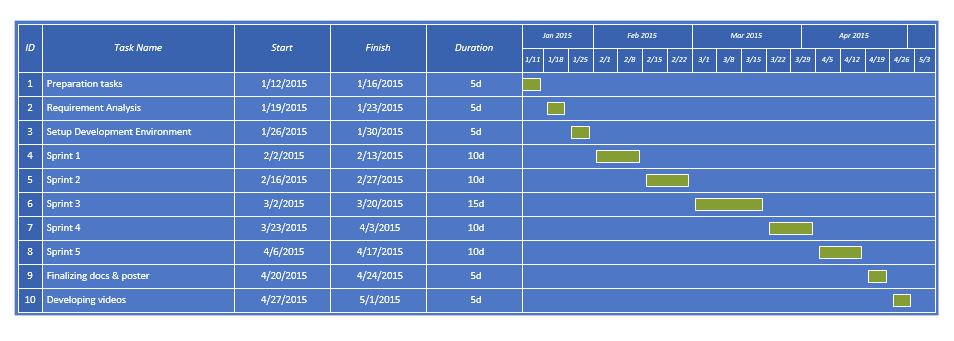
Project management meetings and tasks are generally defined by Mingle, a tool to aid in this agile methodology approach new this semester. Refer to picture below.

1. Appendix

*Senior Project Website V5*



* 1. Appendix A – Project Schedule

Taken from the given syllabus and adapted to what actually occurred.

Gantt chart

### Schedule

|  |  |  |
| --- | --- | --- |
| Week | Date | Topics |
| 1 | 1/12/15 | Project Assignment and Grouping |
| 2 | 1/19/15 | Design & Goal Refinement |
| 3 | 1/26/15 | Design & Goal Refinement & Debugging & Code Refinement |
| 4 | 2/2/15 | Design & Goal Refinement |
| 5 | 2/9/15 | Design & Goal Refinement |
| 6 | 2/16/15 | Design & Goal Refinement |
| 7 | 2/23/15 | Design & Goal Refinement |
| 8 | 3/2/15 | Design & Goal Refinement |
| 9 |  | Spring Break |
| 10 | 3/16/15 | Debugging & Code Refinement |
| 11 | 3/23/15 | Design & Goal Refinement |
| 12 | 3/30/15 | Design & Goal Refinement |
| 13 | 4/6/15 | Design & Goal Refinement |
| 14 | 4/13/15 | Design & Goal Refinement |
| 15 | 4/20/15 | Debugging & Code Refinement & Deliverables |
| 16 | 4/27/15 | Final Presentation & [Demo](https://moodle.cis.fiu.edu/v2.1/mod/assignment/view.php?id=20135) & Poster & Approval Test |

* 1. Appendix B – Feasibility Matrix

Discussion of the following matrix was described in section 2.4.3.

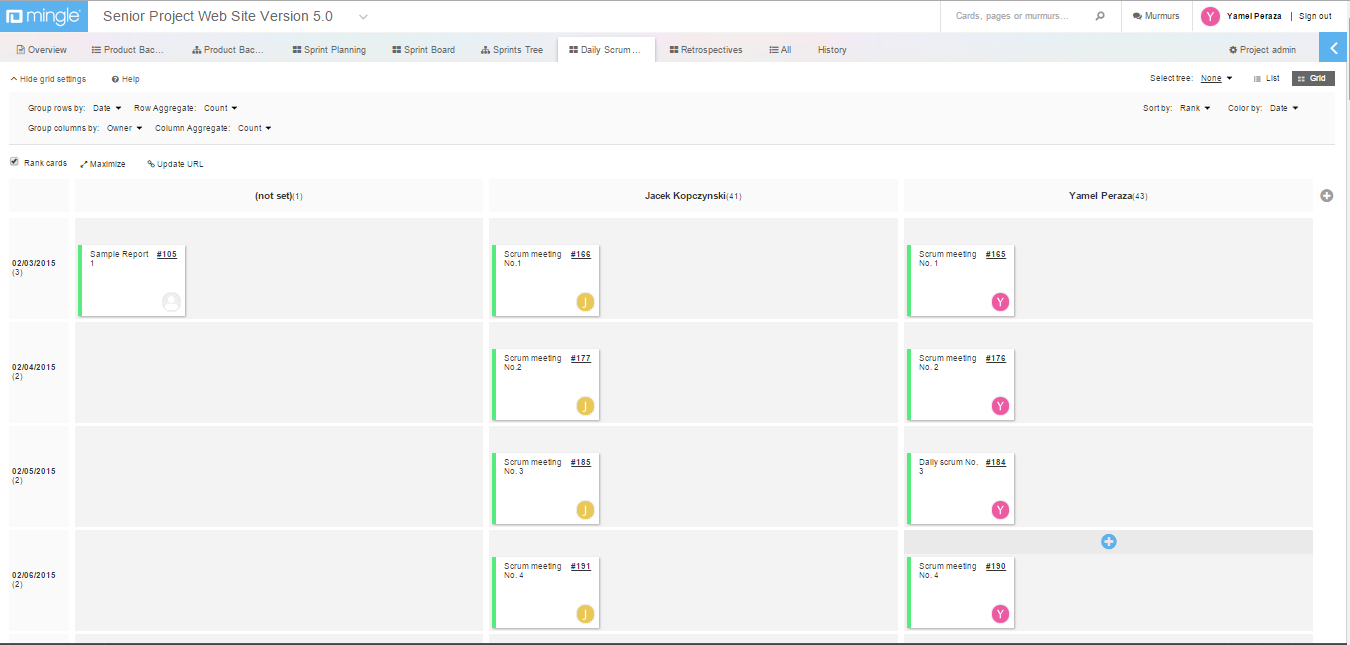
|  |  |  |
| --- | --- | --- |
|  | Alternative I Current System | Alternative II Proposed solution |
| Economic Feasibility | 0 | 0 |
| Technical Feasibility | 0 | 8 |
| Operational Feasibility | 8 | 8 |
| Schedule Feasibility | 10 | 9 |
| Total | 18 | 25 |

* 1. Appendix C – Cost Matrix

|  |  |  |
| --- | --- | --- |
| Project Component | Description – Cost would not be an issue because students are improving on an existing solution to the Senior Project course student, professor, and mentor interaction | % of Total Project Cost |
| Project Management | 0 (Students are working all aspects of the project) | 0 |
| Hardware | 0 (already in place) | 0 |
| Software | 0 (open source) | 0 |
| Testing | 0 (testing tools open source/temporary free license products) | 0 |
| Training | 0 (Students are learning on their own) | 0 |
| Risk Management | 0 (No risk management budget is allocated) | 0 |
| Total | 0 | 0 |

* 1. Appendix D – Diary of Meetings

Developer members of the team met on a daily basis. Below is a screenshot of our daily meeting using Mingle. Meetings with the product owner and lead developer occurred on a weekly basis resulting in the proposed implementation generally described in this document.



1. References

*Senior Project Website V5*



Julio Lopez. "*Feasibility Study & Project Plan for Senior Project Website: Version 4*" (Summer 2014). Florida International University.

Christopher Kerrutt, William Marquez, Cynthia Tope. "*Feasibility Study & Project Plan for Senior Project Website: Version 3*" (Spring 2014). Florida International University.

Fernandez, Yaneli, Camilo Sanchez, and Keiser Moya. "*Feasibility Study & Project Plan for Senior Project Website: Version 1.*" (Spring 2013). Florida International University.

National Resident Matching Program “The Match” Retrieved January 25th 2014, from http://www.nrmp.org/match-process/match-algorithm/

COMBINATORIAL EXPLOSION. (n.d.). Retrieved July 19, 2014, from Web Dictionary of Cybernetics and Systems: http://pespmc1.vub.ac.be/ASC/Combin\_explo.html

Arbelaez, A., Hamadi, Y., & Sebag, M. (n.d.). Online Heuristic Selection in Constraint Programming. Retrieved July 07, 2014, from research microsoft: http://research.microsoft.com/en-us/people/youssefh/search-socs.pdf