OneTeam **Software Design Document  
μProject**short line

**A micro project management tool**

David Blair

Christian Carducci

Juan Landaverde

Sang-Joon Lee

Jacqueline Pham

Srivathsa Rajagopal

August 1, 2016

# **Table of Contents**

[Table of Contents](#h.qezgdvhmuoh)

[Section A. μProject Background](#h.gexkng1roy73)

[Project Overview](#h.pnc9pd3mcelk)

[Target Users](#h.4ceto6yrkxk4)

[Section B. Design Overview](#h.vx23hjh28tvo)

[B.1 Description of Problem](#h.hkd65nh3y13n)

[B.2 Technology Used](#h.2clu617hsu54)

[B.3 System Architecture](#h.yixg7i9vrin7)

[B.4 System Operation](#h.u4zxlw37zlja)

[Section C. Functional Design](#h.ogpq04k59bwi)

[C.1. μProject Software Design](#h.rj32jot74h5k)

[User Login & Authentication](#h.qr3fxnwx566z)

[C.2. Queued: Project Management Tool Functional Design](#h.isqa1xdgyyr8)

[C.2.1. Queued Database Design](#h.jd1jh8x69ltx)

[C.2.2. “Must-Haves” Functional Detail Design](#h.vaw6owh2asr4)

[C.2.3. “Wants” Functional Detail Design](#h.ufb662xmjclc)

[C.2.4. “Nice-to-Haves” Functional Detail Design](#h.t7f8k4eh2sex)

[C.3. ChatNow: Chat System Functional Design](#h.dpg3lxxph8tp)

[C.3.1. ChatNow Database Design](#h.dnotmn4ia1ky)

[C.3.2. “Must-Haves” Functional Detail Design](#h.69ob4u2jwkmc)

[C.3.3. “Wants” Functional Detail Design](#h.gvy01k1isbg2)

[C.3.4. “Nice-to-Haves” Functional Detail Design](#h.iye3h4btl1ou)

[C.4. Bugs: Issue Tracker Functional Design](#h.payejds7i9sr)

[C.4.1. Bugs Database Design](#h.8orl7097q9iw)

[C.4.2. “Must-Haves” Functional Detail Design](#h.ig0sldpq2ms0)

[C.4.3. “Wants” Functional Detail Design](#h.fzdjdq99vca4)

[C.4.4. Nice-to-Haves Functional Detail Design](#h.afzco0lch4pj)

[Section D. User Interface Design](#h.t9v01w7clqr9)

[D.1. uProject Home Page UI Design](#h.mxvkhtho6yo8)

[Project Home Page UI](#h.g77nmsqk211t)

[Project Settings UI](#h.m02xqixtxccg)

[D.2. Queued UI Design](#h.fxqje0gf4i0e)

[Queued Home Page UI](#h.ve3ob3aivb3s)

[Queued Add Project UI Layout](#h.1zlcfyi5ejs0)

[Queued Create Project UI Layout](#h.jtovzeogo74e)

[Queued Story View UI Layout](#h.j117xm5r9pyx)

[Queued Add Story UI Layout](#h.slgzgjirarsj)

[D.3. ChatNow UI Design](#h.op670jr8rb4i)

[ChatNow Integrated UI Layout](#h.xj7vrjjke9nv)

[D.4. Bugs UI Design](#h.d4nyimkd6m9j)

[Bugs Home Page UI Layout](#h.ei7rj0o6leog)

[Bugs Create Issue UI Layout](#h.vrog6ccib3dg)

[Bugs Detail View UI Layout](#h.nu7mjccky6bw)

[Bugs Archive View UI Layout](#h.feaan1nwdsqn)

[Section E. References](#h.bktzezggvzl3)

# 

# 

# **Section A. μProject Background**

## **Project Overview**

*Purpose of Application:*

To provide capabilities to support a software project using Agile methodology through a basic project management tool.

*Major Functionalities:*

* Management Component - manages the project's requirements
* Communication Component - allows project team to communicate via real time chat
* Issue Management Component - manages the project’s issues or defects

## **Target Users**

The target demographic for this application will be small to medium sized (1 to 8 people) software teams using Agile methodology who need a basic tool to support the process.

# 

# **Section B.** Design Overview

# B.1 Description of Problem

To provide project management web application with capabilities to support a software project using Agile methodology through a basic project management tool.

# B.2 Technology Used

The application (μProject) is comprised of three relatively separate components wrapped in a single web application. The elements are described below as three separate projects. The descriptions outline the project team, goal of the project, and address the background, description of activities, and our proposed activities by iteration. We propose to build all three components using Node.js as the foundation for the backend, with MySQL as the database server, and Bootstrap as the front-end framework.

# B.3 System Architecture

This project is implemented using Model View Controller architecture. The following describes technologies used for Model View Controller.

* Model View Controller
  + **Model**
    - MySql DB
  + **View** - templates
    - HandlebarsJS
    - Bootstrap
    - jQuery, Various small libraries
  + **Controller**
    - NodeJS, ExpressJS, SocketIO

# B.4 System Operation

The uProject application is deployed on Amazon Web Server on following URL:

<http://uproject.thirdelement.com/>

# B.6 Software Application Source Code

The uProject application is an open project and all source code is stored on github. You can find source code on following link:

<https://github.com/CS673Smr16Team1/project>

# **Section C. Functional Design**

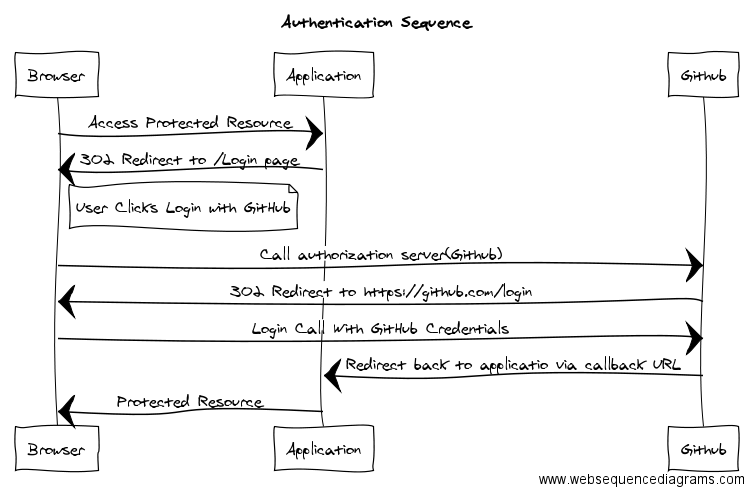
# C.1. μProject Software Design

< add functional design description for user login, authentication, dashboard functionalities, setup page, etc >

## **User Login & Authentication**

The authentication is performed using github account. The following diagram illustrate sequence of timing diagram to authenticate users using github account. Since all users have access to github account and can easily obtain github account, this application can safely assume that user can successfully log-in using github account.

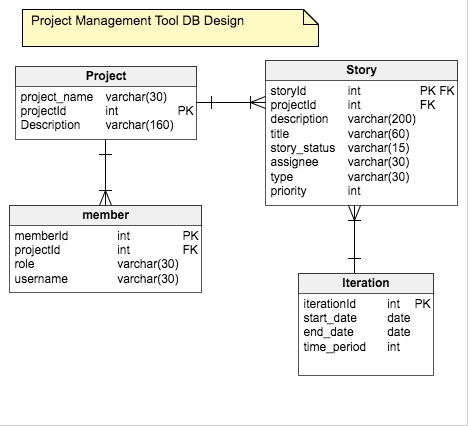
*Sequence Diagram: Authentication*



# **C.**2**. Queued: Project Management Tool Functional Design**

|  |  |
| --- | --- |
| Functional Description | To create a project requirements management tool for Agile software development projects which allows user to create a project, stories and manage lifecycle of software development tasks. |

## **C.2.1. Queued Database Design**



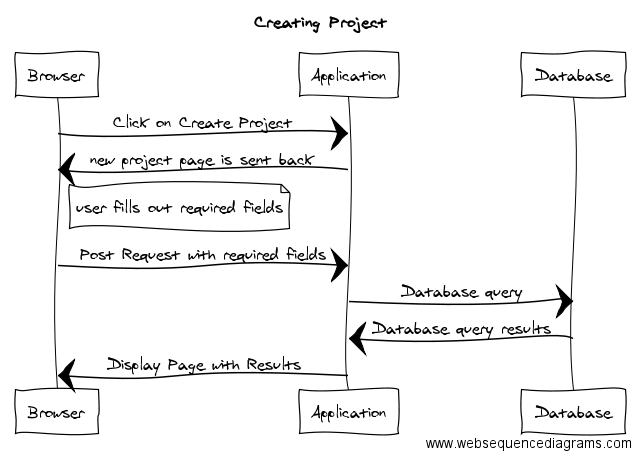
## **C.2.2. “Must-Haves” Functional Detail Design**

*Create a user*

The authentication is performed using github account.

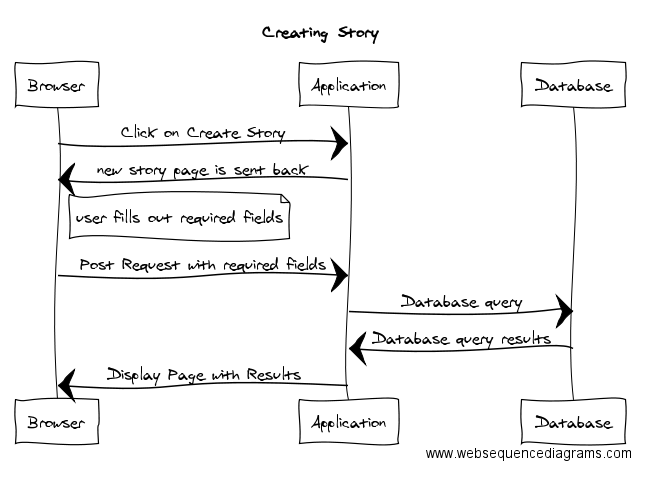
*Create a Project: main project with multiple individuals/users*

The following is sequence diagram that describe flow of creating a project.



*Create, edit, delete Stories: small pieces of work or features*

The following is sequence diagram that describe flow of creating a story.



*Ability to assign a story to a user for development*

Ability to assign and update status of a story throughout development cycle

*Backlog of Stories: list of prioritized stories that can be worked on, ability to re-order stories*

*Current Stories: list of prioritized stories that are currently under development*

*Completed Stories: list of stories that are completed*

## **C.2.3. “Wants” Functional Detail Design**

Assign type of story (feature, bug or release)

## **C.2.4. “Nice-to-Haves” Functional Detail Design**

*Create and manage multiple projects: ability to create, edit and delete multiple projects*

*View of user owned stories*

*Email reminders for Story deadlines*

*Archive Projects and Stories*

*Scroll bar for lists*

# 

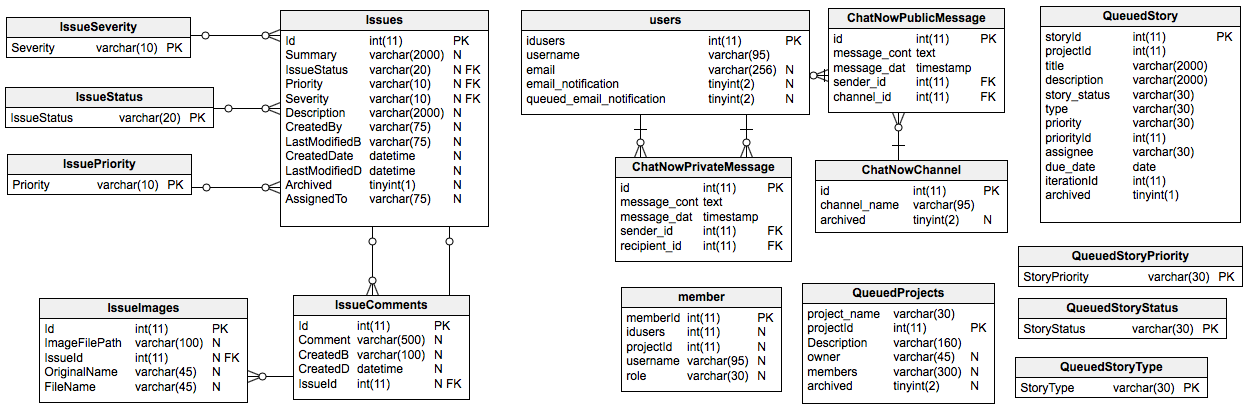
# 

# **C.**3**. ChatNow: Chat System Functional Design**

|  |  |
| --- | --- |
| Design Description | An easy-to-use communication environment that allows a team to share ideas in a virtual chat room setting as an alternative to emails. |

## **C.3.1. ChatNow Database Design**

The following is Database Design for ChatNow application.



## **C.3.2. “Must-Haves” Functional Detail Design**

*Create a channel - ability to create a public chat room*

# index.png

*Archive chat history - make chat history persistent (not session based, but always visible)*

## 

## **C.3.3. “Wants” Functional Detail Design**

*Send private messages to other members*

*Search - ability to search through the chat history for keywords*

## 

## **C.3.4. “Nice-to-Haves” Functional Detail Design**

*Email Notifications*

*Archive/unarchive a channel*

*Online Status of other users*

*New Private Message Notification*

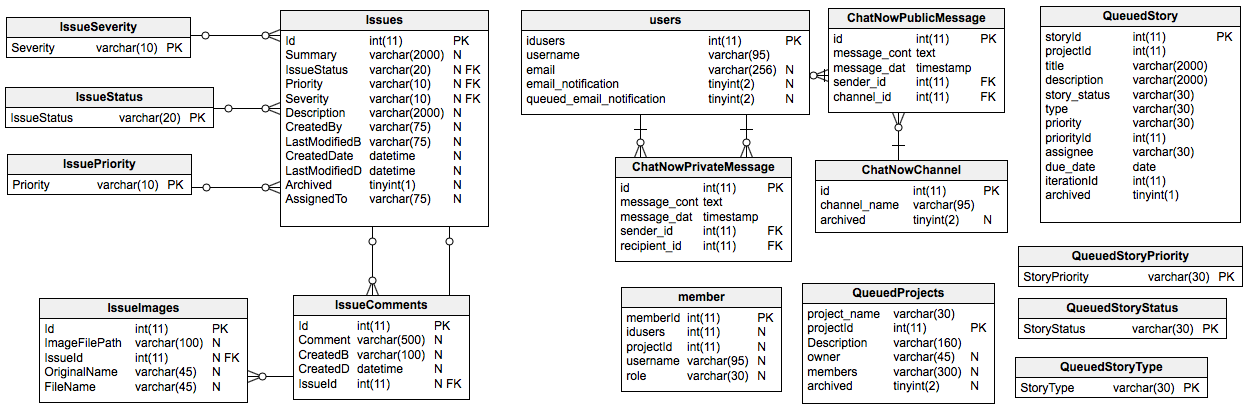
*Infinite Scrolling*

# **C.**4**. Bugs: Issue Tracker Functional Design**

|  |  |
| --- | --- |
| Functional Description | A basic tracking system to create, assign, update and resolve issues. |

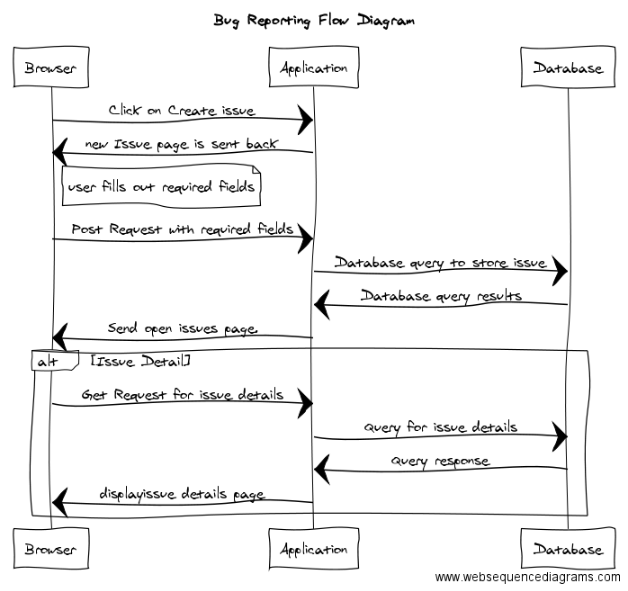
## **C.4.1. Bugs Database Design**

The following is database design for Bugs issue tracking application.



## **C.4.2. “Must-Haves” Functional Detail Design**

*Create, edit, update, and delete issues*



*Set priority, severity and status of issue*

*Identify who reported issue*

## **C.4.3. “Wants” Functional Detail Design**

*Comment on issue*

*Assign issue responsibility to user*

*Attach screenshots to issue*

## **C.4.4. Nice-to-Haves Functional Detail Design**

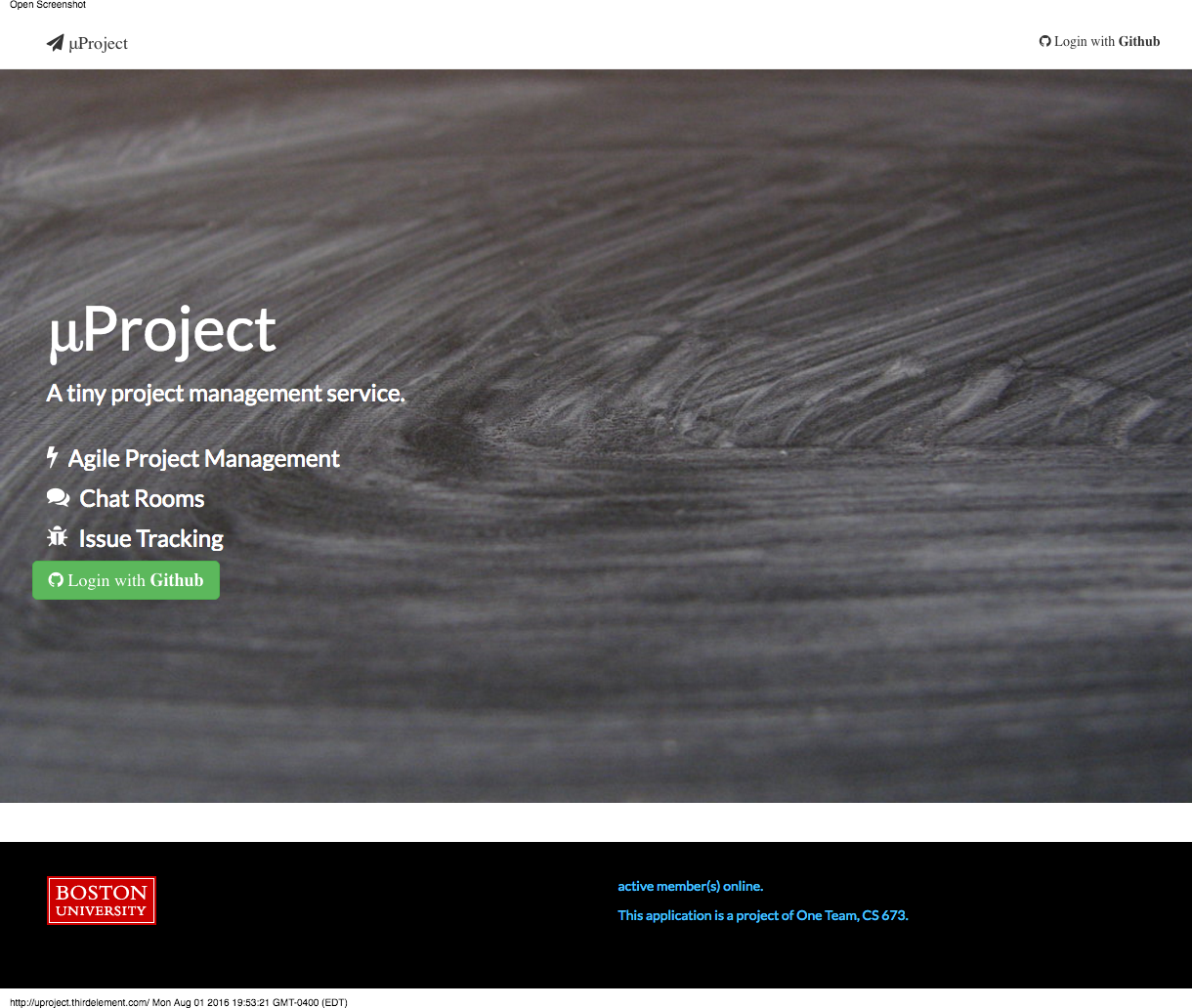
*Filtering of Issue List*

# 

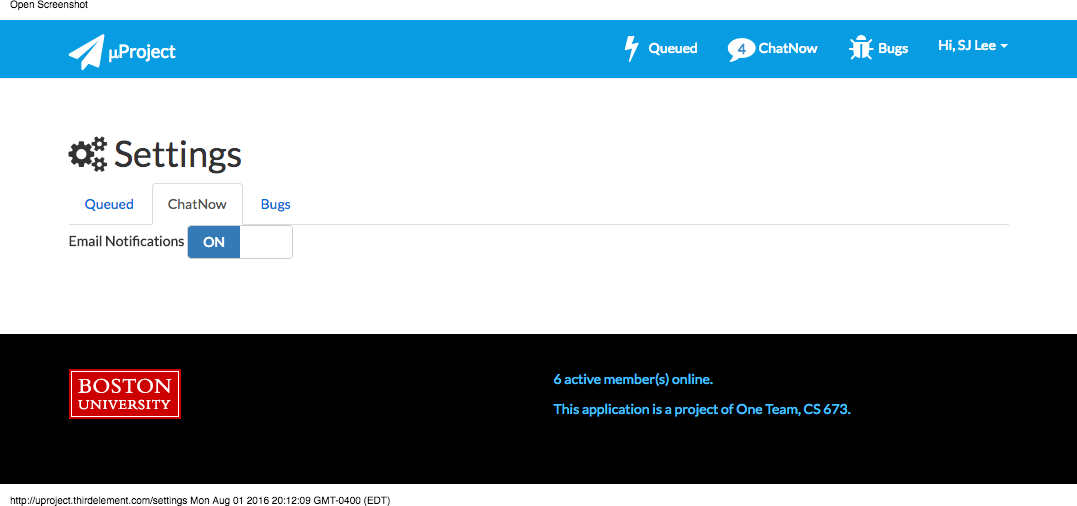
# **Section D. User Interface Design**

# D.1. uProject Home Page UI Design

## **Project Home Page UI**

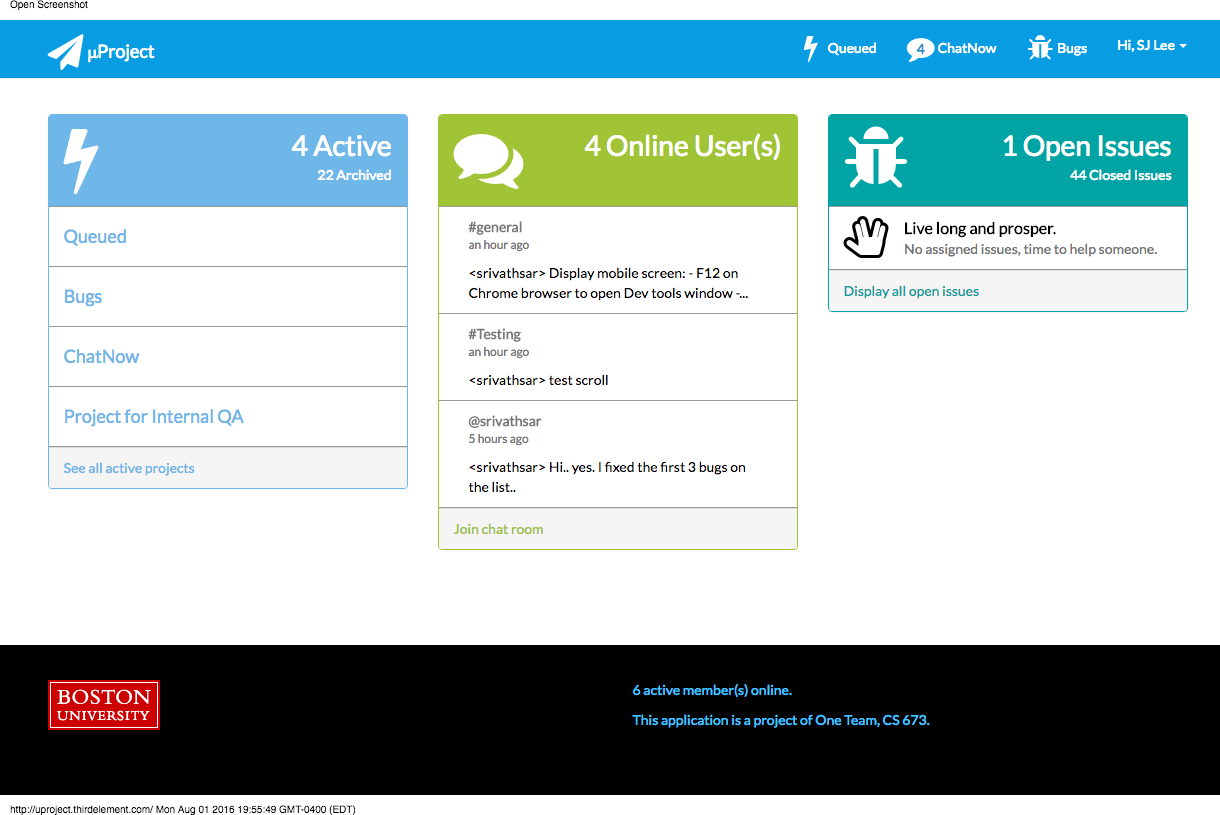


## **Project Settings UI**

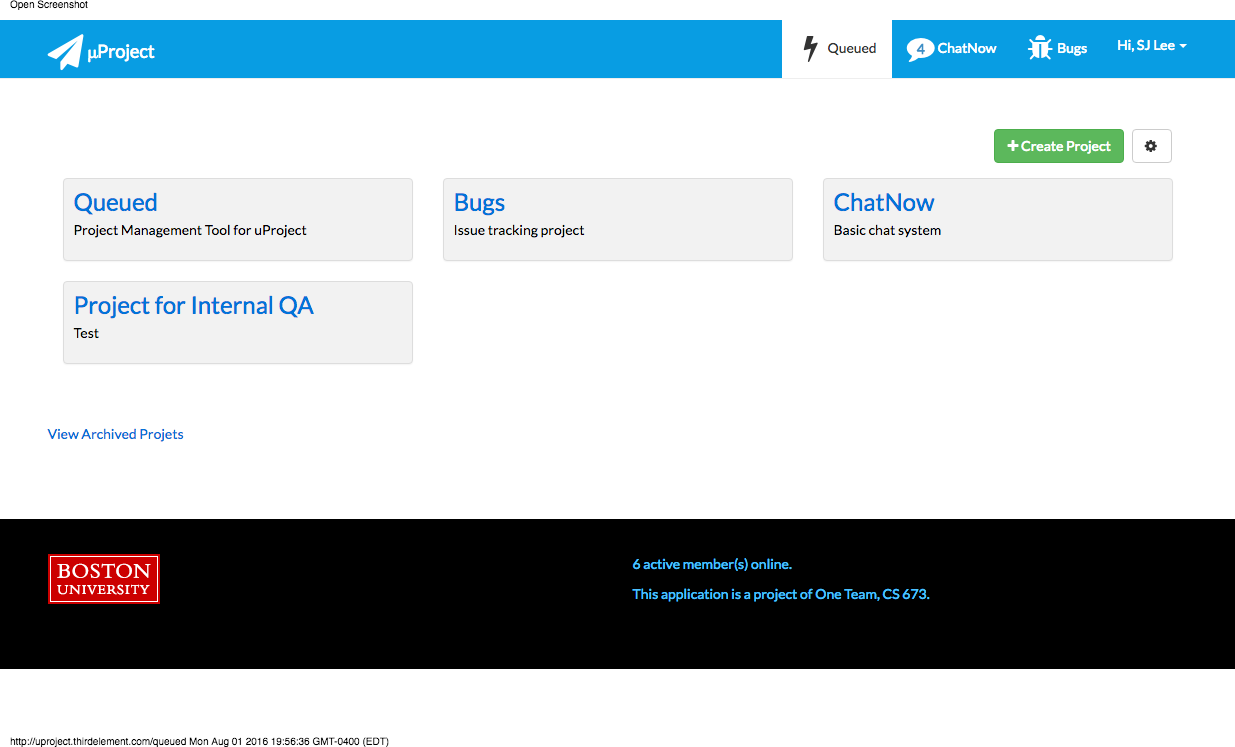


# D.2. Queued UI Design

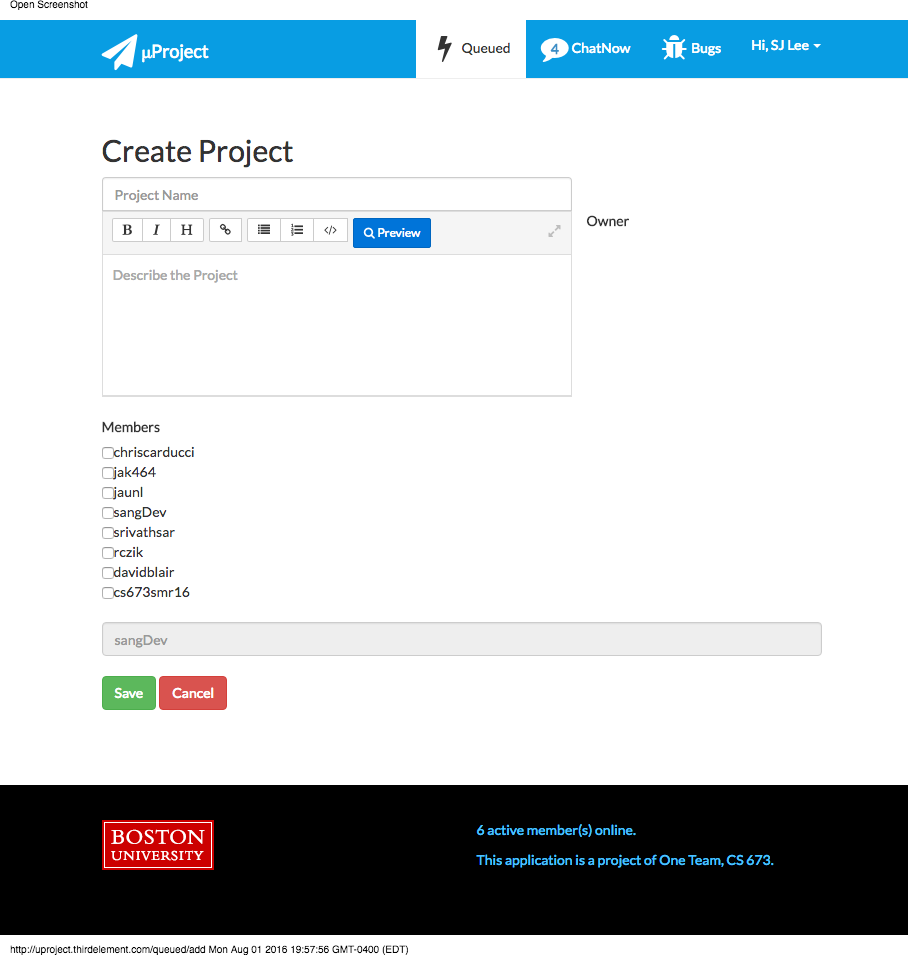
## **Queued Home Page UI**



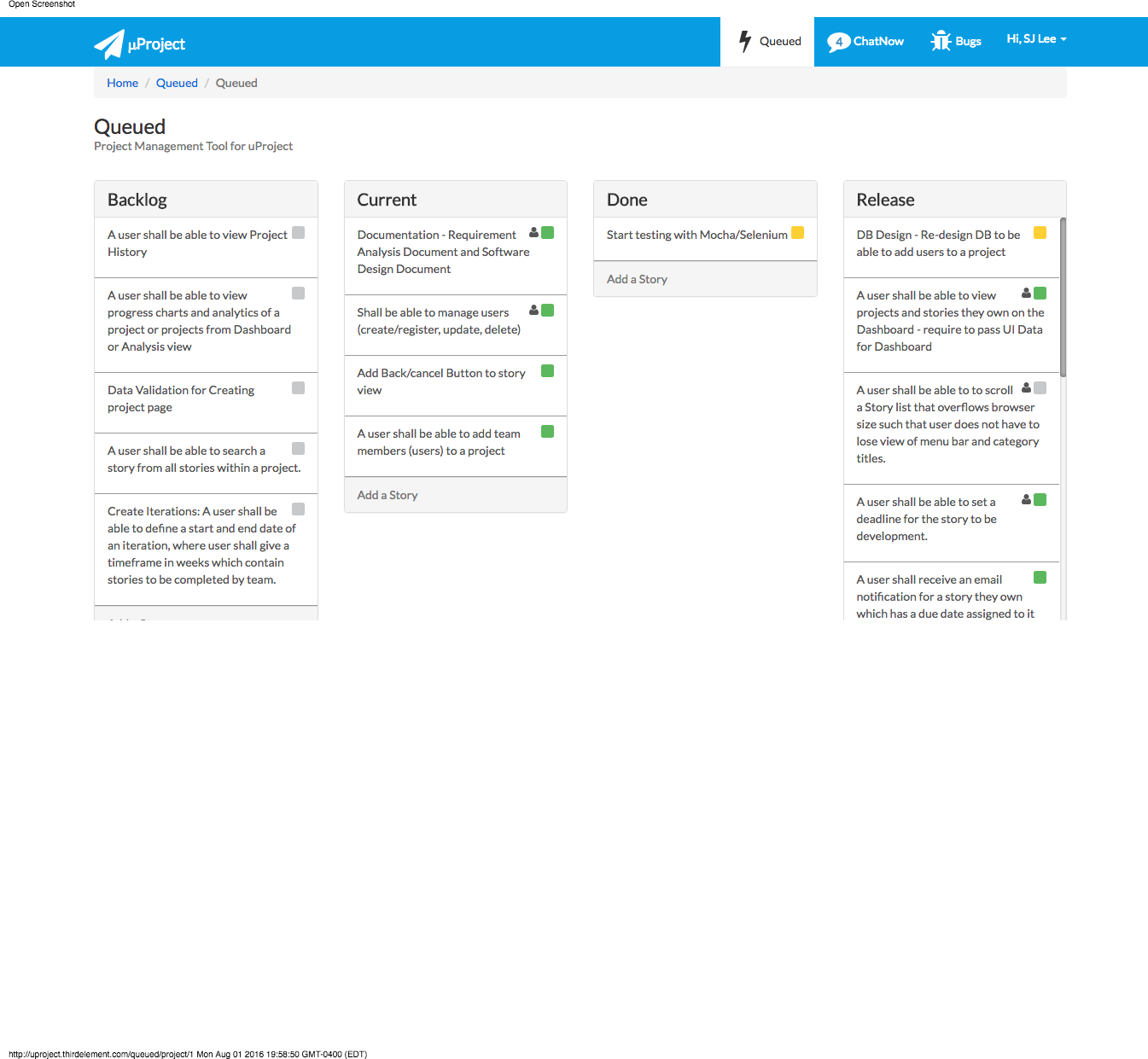
## **Queued Add Project UI Layout**



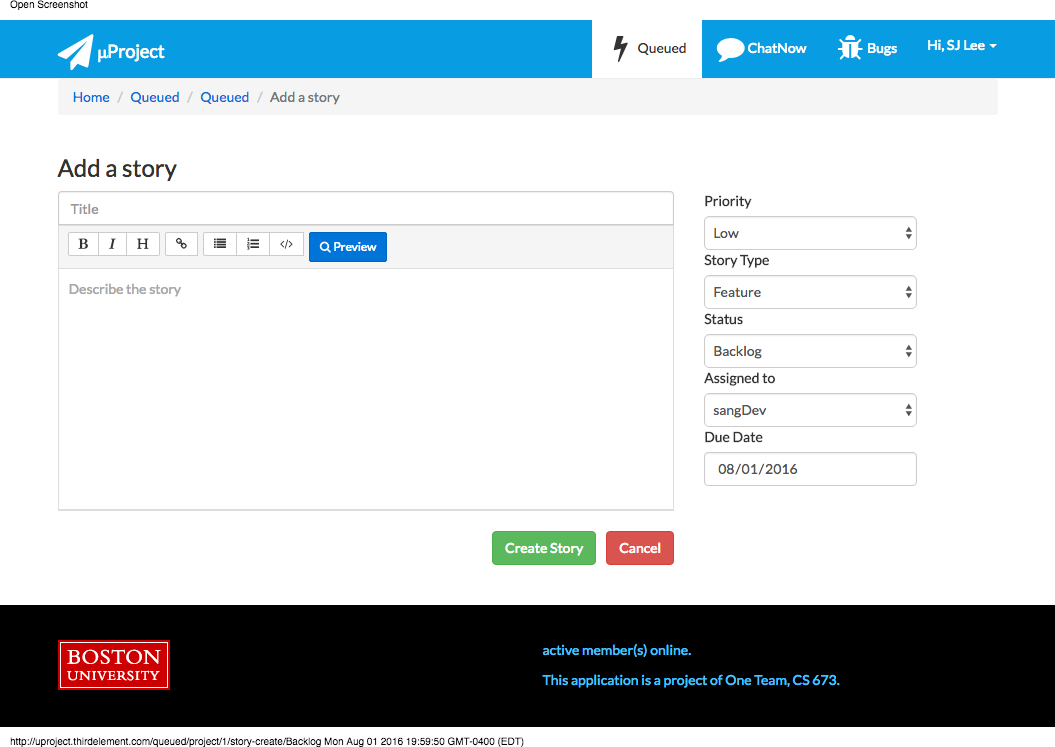
## **Queued Create Project UI Layout**



## **Queued Story View UI Layout**

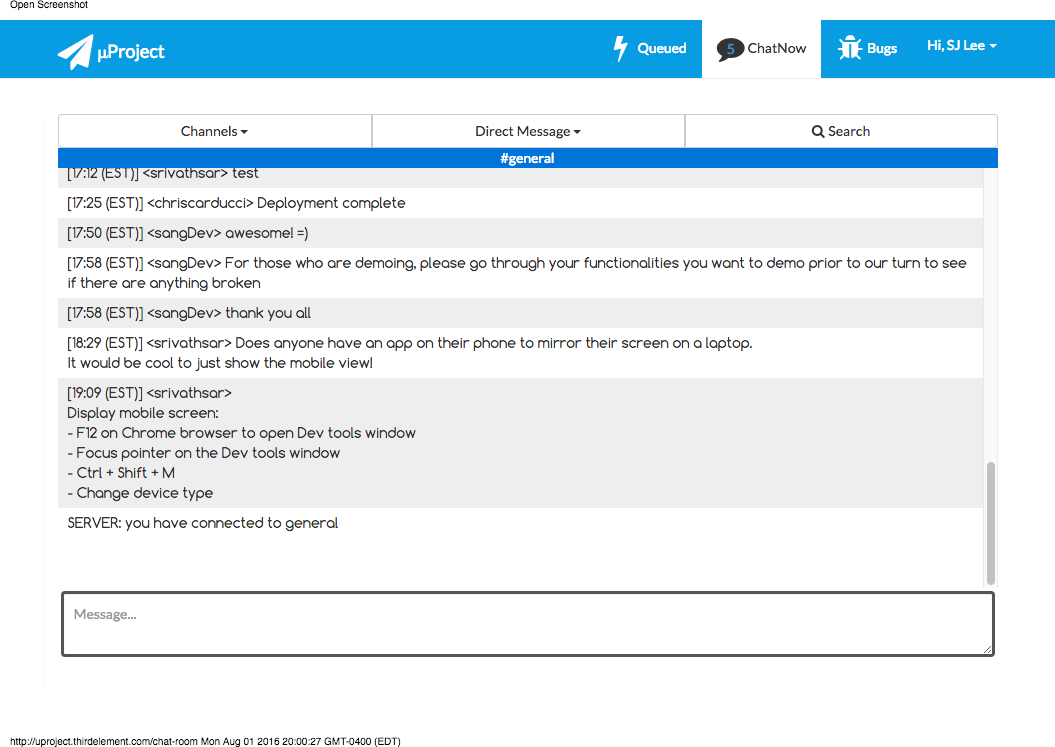


## **Queued Add Story UI Layout**



# D.3. ChatNow UI Design

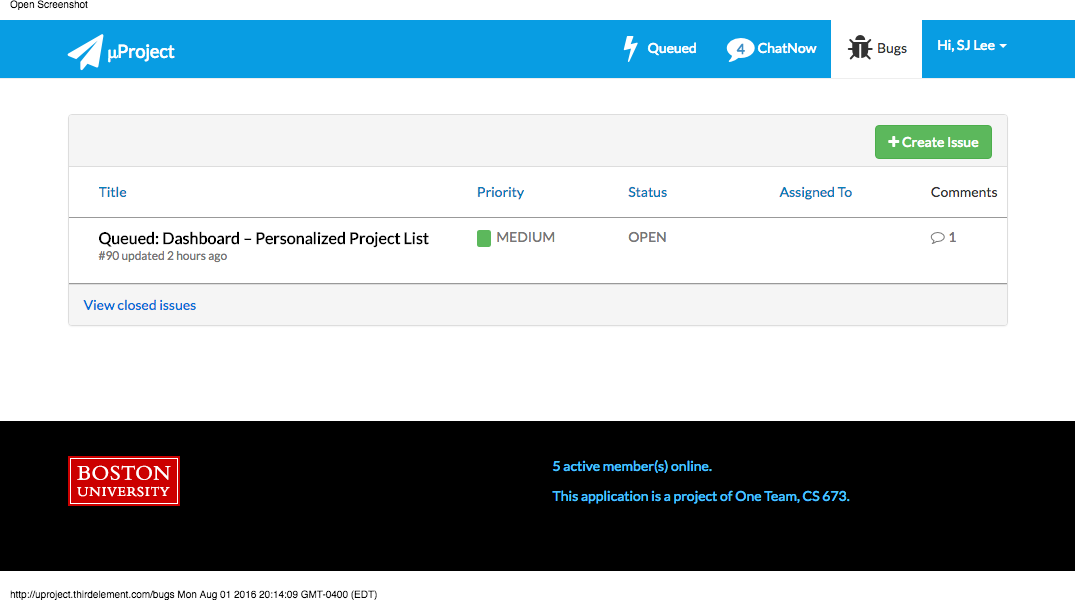
## **ChatNow Integrated UI Layout**



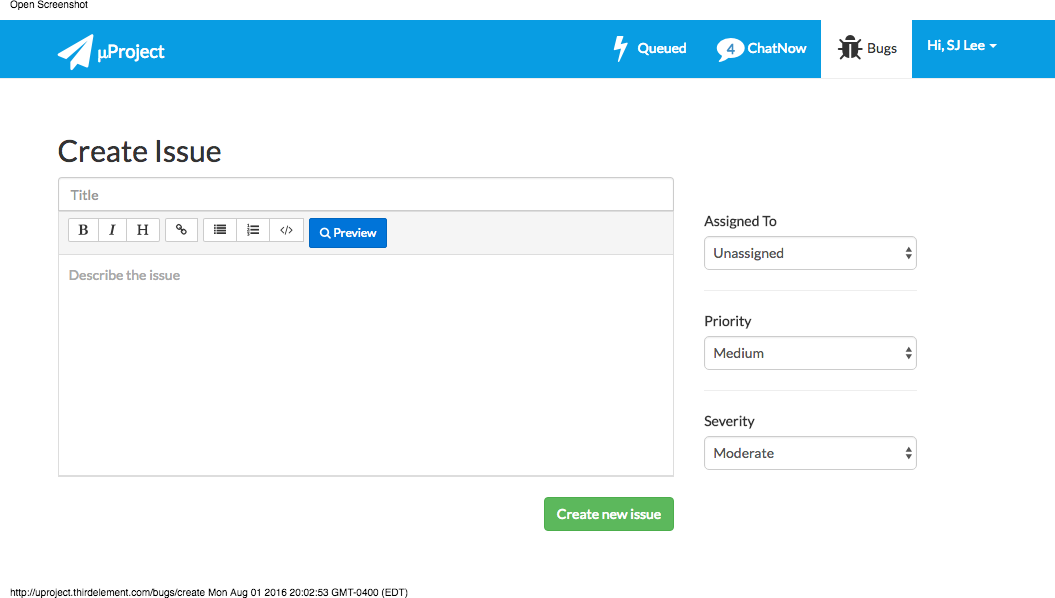
# 

# D.4. Bugs UI Design

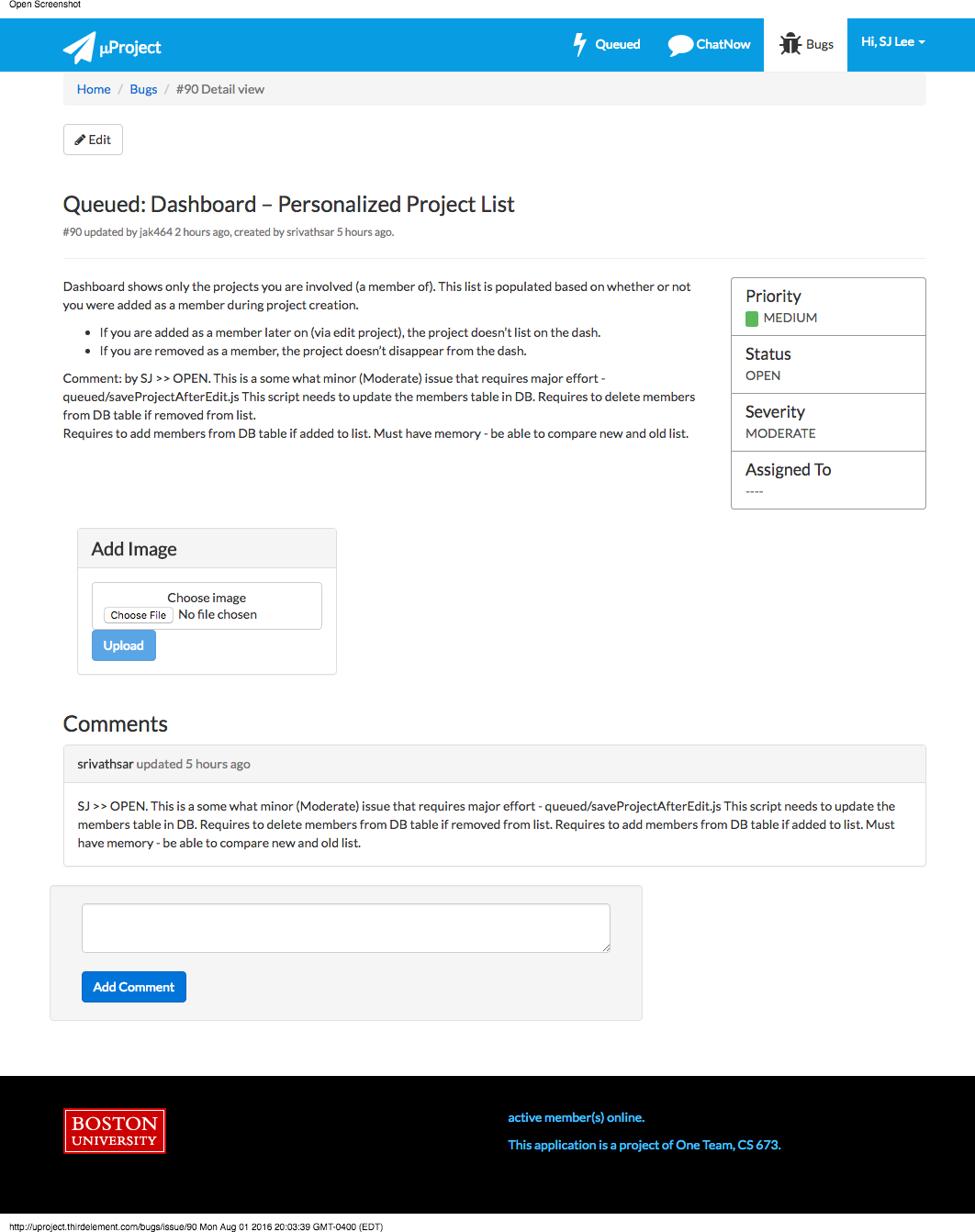
## **Bugs Home Page UI Layout**



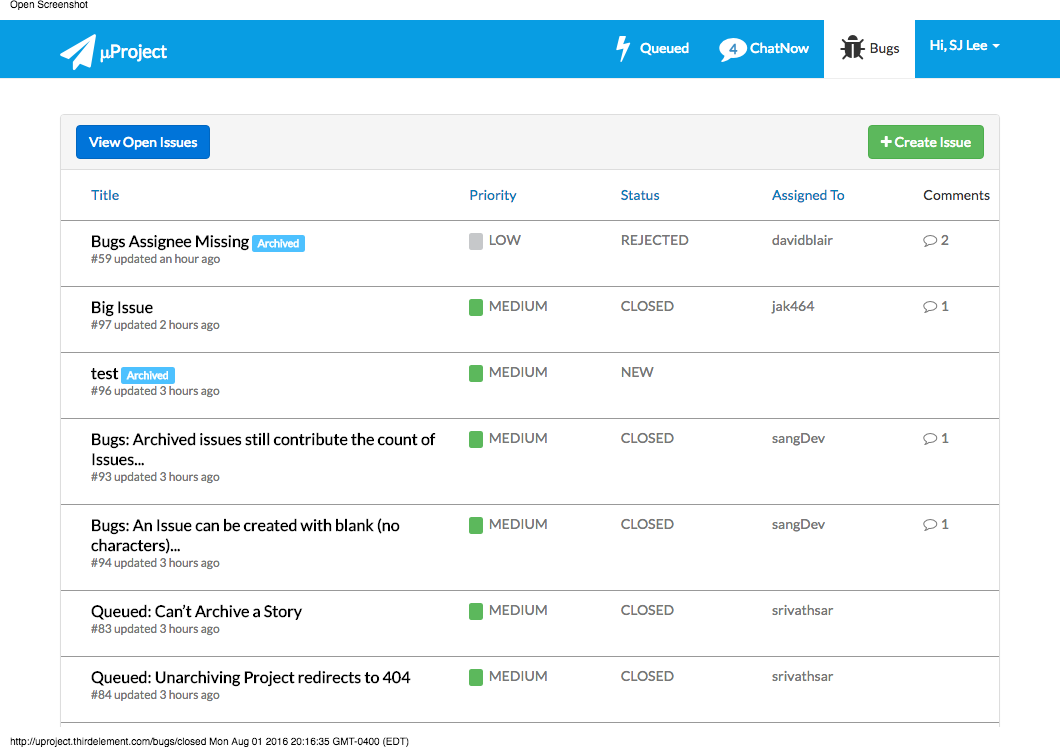
## **Bugs Create Issue UI Layout**



## **Bugs Detail View UI Layout**



## **Bugs Archive View UI Layout**



# 

# 

# **Section E.** References

* Node.js : <https://nodejs.org/en/>
* Express.js: <https://expressjs.com/>
* Mocha: <https://mochajs.org/>
* Chai: <http://chaijs.com/>
* Selenium: <http://www.seleniumhq.org/>
* WebStrom: <https://www.jetbrains.com/webstorm/>