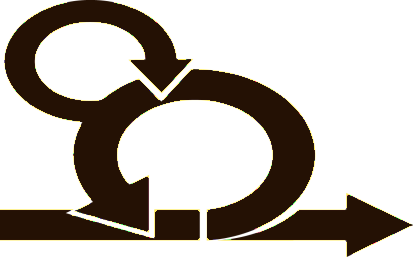
Final Year Project

(Advisor: Sohail Imran)

#### **Game plan 2**



##### Software PLANNING with ***Scrum*** Framework

Anees Ahmed (58875) - MCS

Table of Contents

[**Table of Figures** 3](#_Toc487736209)

[An Overview of Software Planning with Scrum Framework 4](#_Toc487736210)

[Scrum Flow 5](#_Toc487736211)

[The Synopsis of **GAMEPLAN 2** 7](#_Toc487736212)

[Portfolio: The Project Dashboard Board 8](#_Toc487736213)

[Releases: working starts here 10](#_Toc487736214)

[User STORIES: Requirement analysis and product backlog 11](#_Toc487736215)

[sprint planning: choosing user stories and plan development 13](#_Toc487736216)

[sprint Execution: Peform DEVELOPMENT, test, review and log task against user stories 14](#_Toc487736217)

[Core Technolgies 15](#_Toc487736218)

[Key Functions 16](#_Toc487736219)

[User Management in gameplan 2 17](#_Toc487736220)

[Entity Relationship Diagram 19](#_Toc487736221)

# **Table of Figures**

[Figure 1:short incremental development cycle](file:///Z:\activelearning\gameplan2\presentation\project report.docx" \l "_Toc487736098) 4

[Figure 2: the Scrum Process flow](file:///Z:\activelearning\gameplan2\presentation\project report.docx" \l "_Toc487736099) 6

[Figure 3: Portfolio project dashboard 8](#_Toc487736100)

[Figure 4: Data entry form](file:///Z:\activelearning\gameplan2\presentation\project report.docx" \l "_Toc487736101) 8

[Figure 5: Release Form](file:///Z:\activelearning\gameplan2\presentation\project report.docx" \l "_Toc487736102) 10

[Figure 6:Releases](file:///Z:\activelearning\gameplan2\presentation\project report.docx" \l "_Toc487736103) 11

[Figure 7: user management 18](#_Toc487736104)

[Figure 8: Er diagram](file:///Z:\activelearning\gameplan2\presentation\project report.docx" \l "_Toc487736105) 19

# An Overview of Software Planning with Scrum Framework

**GAMEPLAN 2** is a software planning tool. It helps development and infrastructure planners to prepare their plans and execute them using scrum framework. It is based on **scrum** framework , famous in software development and support projects.

Scrum is a project management framework. It is lightweight in nature, but due to its well structured design, it could be used for large scale of diversify projects.

In scrum we follow iterative and incremental approach for planning and executing our software projects or products. Increments are smaller in sizer and each iteration takes less time (1 to two months) than traditional project management practices. Less time means more short time release and stack holder could have better understanding of new changes, fixes and improvement.



Figure 1:short incremental development cycle

# Scrum Flow

|  |  |  |
| --- | --- | --- |
| 1 | Development initiative | Initial preparation like selecting team, budget and determine scope etc. |
| 2 | Framework planning | Scum preparation working like setting sprint duration, product owner(s), scrum master |
| 3 | Requirement analysis ( User stories) | Collecting requirements from product owner in form of user stories and develop a product backlog. This product backlog will be use for upcoming sprint |
| 4 | Sprint Planning | Selecting user stories from the product backlog (unassigned user stories) and preparing a sprint log of stories that will be accomplished and delivered during this release.  Creating Task required to make user stories as complete. |
| 5 | Development (sprint execution) | Performing development, testing and reviewing tasks. Creating further tasks if required. Daily reviewing the progress of sprint (scrum daily meeting) |
| 5 | Release and Review | End of a sprint. All user stories fulfilled by performing required tasks. |

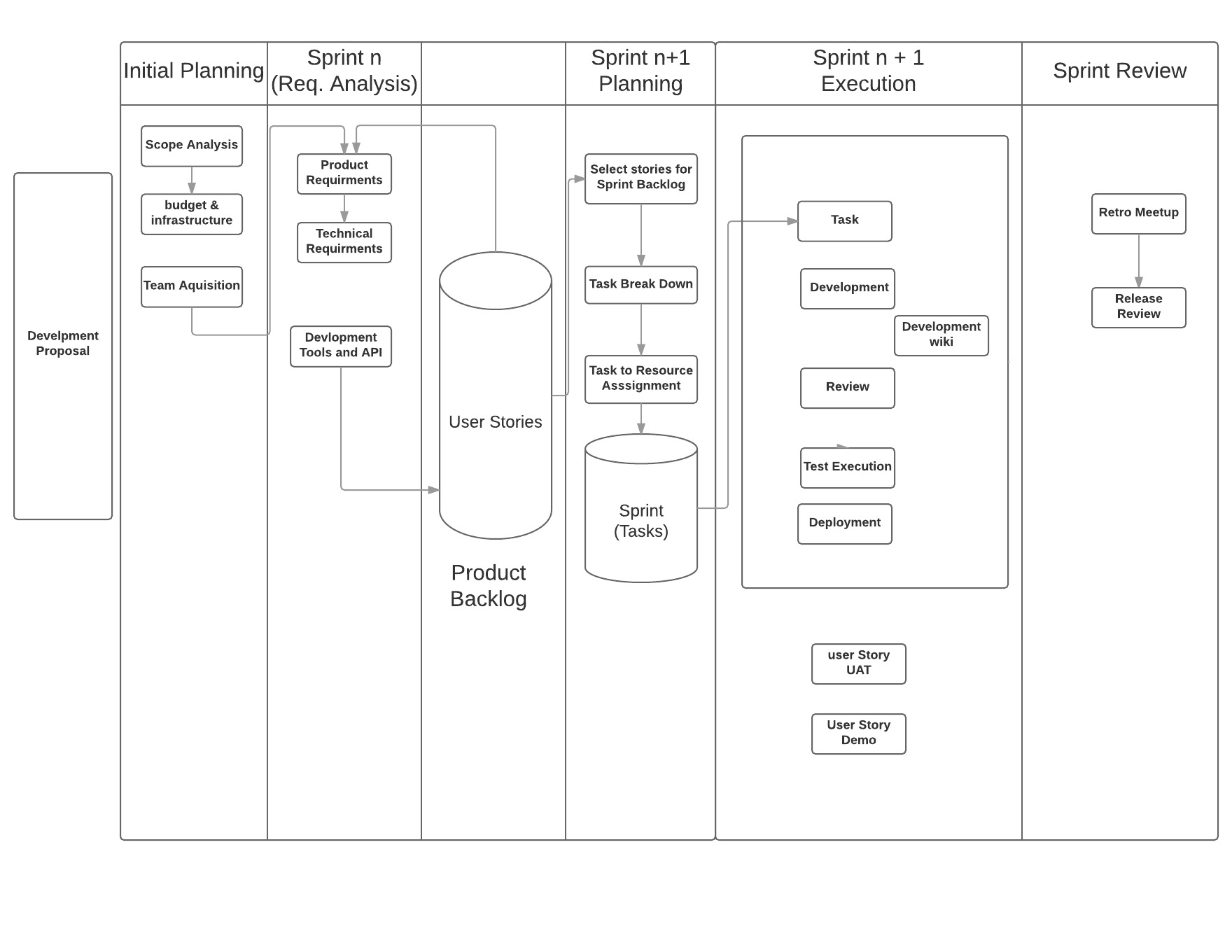


Figure 2: the Scrum Process flow

# The Synopsis of **GAMEPLAN 2**

|  |  |
| --- | --- |
| 1 | Portfolio: The Project Dashboard Board |
| 2 | Releases: Project Working Starts Here |
| 3 | User Stories: Making the Product Backlog |
| 4 | Sprint Planning: Choosing Stories for Development |
| 5 | Sprint Execution: Developing, Testing, Reviewing |
| 7 | Retro Meeting: End of Sprint Analysis |

# Portfolio: The Project Dashboard Board

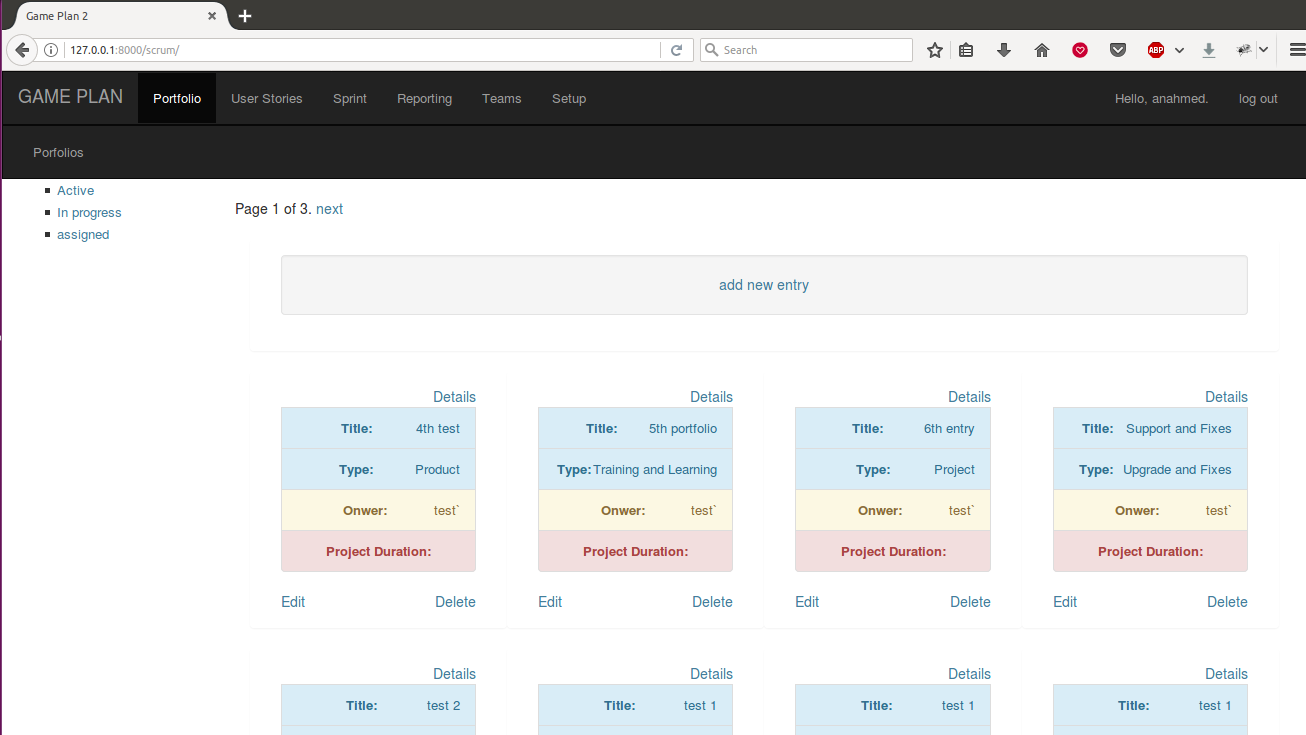


Figure 3: Portfolio project dashboard

portfolio board or project dashboard is starting point for our GAMEPLAN 2 software. The dashboard provide you to get inside the required project or product ( click details). You can edit your project information; change its status or remove, if needed. The *sidebar*  provides a filter to select projects by their current status.

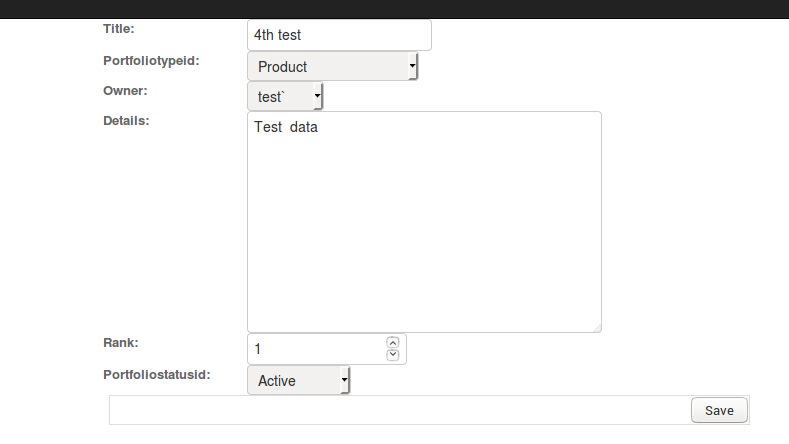


Figure 4: Data entry form

**Design Details:**

The Portfolio page follow a MVT approach. Each url routes call its view where the view class fetch data from model and send to a template for viewing and data entry.

Main url : http://127.0.0.1:8000/scrum

|  |  |  |  |
| --- | --- | --- | --- |
| URL | View | Template | Model |
| **^portfolio**  (Portfolio Dashboard) | PortfolioList | PortfolioList.html | Portfolio |
| **^portfolio/(?P<pk>\d+)$**  (moves to details pages and shows project details and its releases) | PortfolioDetails | PortfolioDetails.html | Portfolio |
| **^portfolio/add** | PortfolioCreate | portfolio\_form.html | Portfolio |
| **^portfolio/update/(?P<pk>\d+)$** | PortfolioUpdate | portfolio\_form.html | Portfolio |
| **^portfolio/delete/(?P<pk>\d+)$** | PortfolioDelete | conform\_delete.html | Portfolio |

# Releases: working starts here

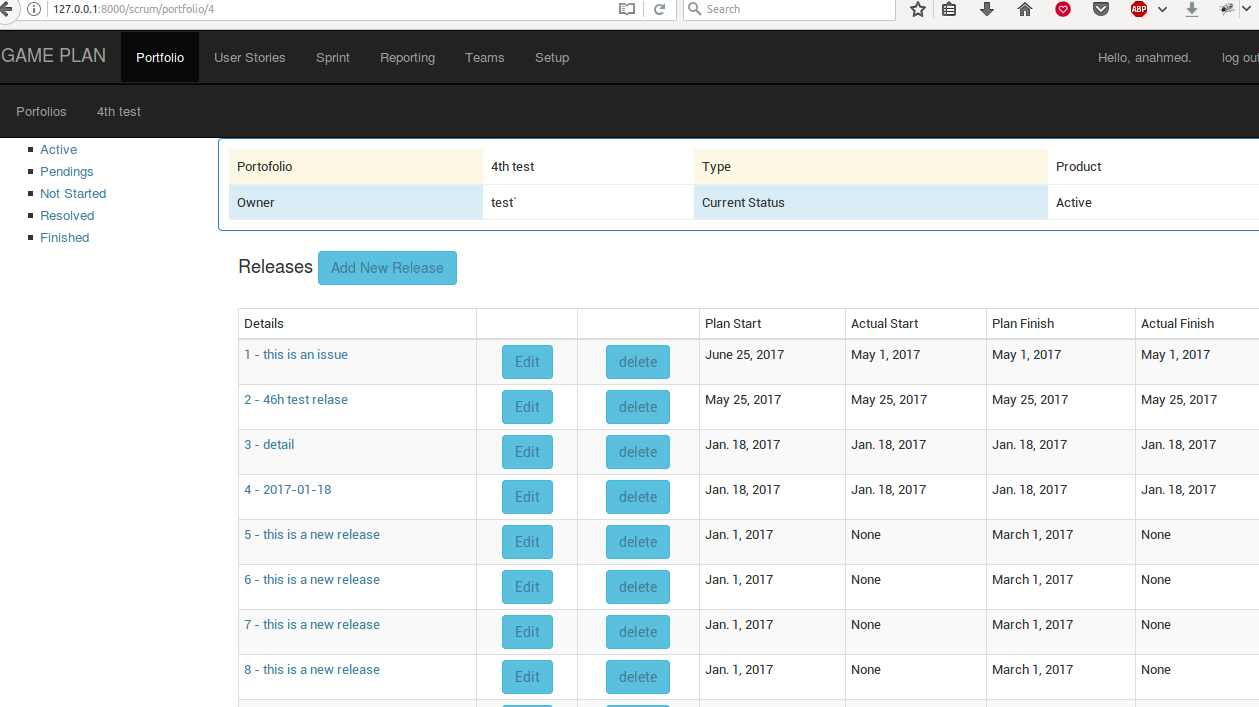


Figure 5: Release Form

In the detail page of porfolio, Project Releases are listed. Choosing any release moves to its user story backlog and sprint listing. For any type of portfolio, at least one release is required. As Gameplan2 follows Scrum methology (increamental, iterative), more release of a single portfolio are common.

**Design Details:**

Main url : http://127.0.0.1:8000/scrum

|  |  |  |  |
| --- | --- | --- | --- |
| URL | View | Template | Model |
| **^release**  (Portfolio details page, showing releases) | ReleaseList | ReleaseList.html | Release |
| **^release/(?P<pk>\d+)$**  (Details of selected release, showing user stories and sprints) | ReleaseDetails | ReleaseDetails.html | Release |
| **^release/add** | ReleaseCreate | release \_form.html | Release |
| **^portfolio/update/(?P<pk>\d+)$** | ReleaseUpdate | release \_form.html | Release |
| **^release/delete/(?P<pk>\d+)$**  Deleting existing release | ReleaseDelete | release \_delete.html | Release |

# User STORIES: Requirement analysis and product backlog

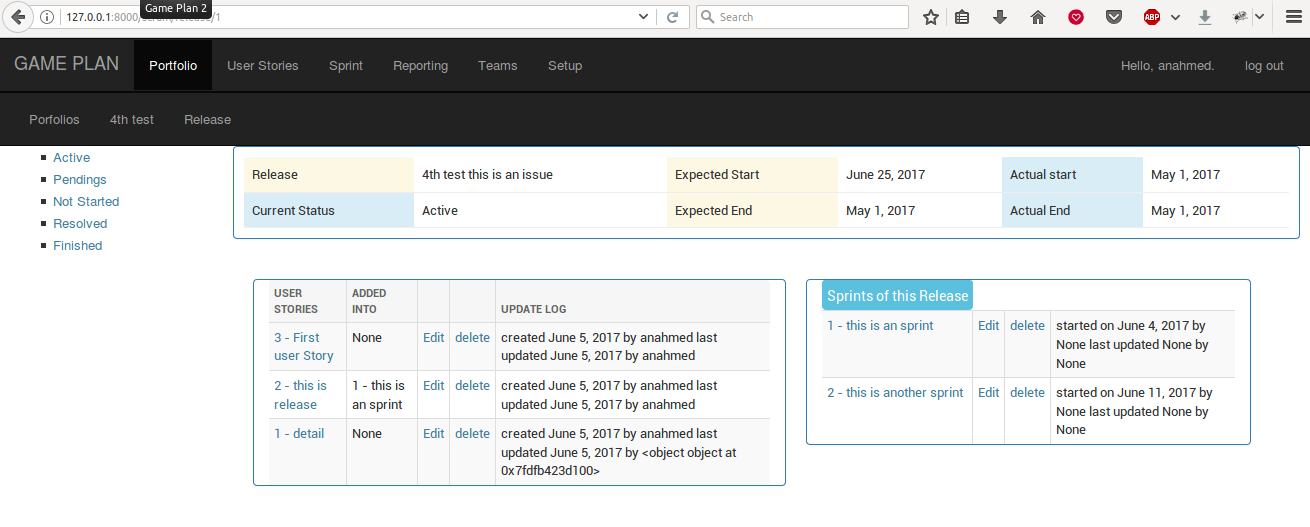
A user story is the heart of scrum model. In scrum, all requirements are stored as user stories. Each release is comprises of a set of user stories that are not assigned to any sprint (product backlog) and different sprints plan. Each sprint plan has selected set of user stories.

Figure 6:Releases

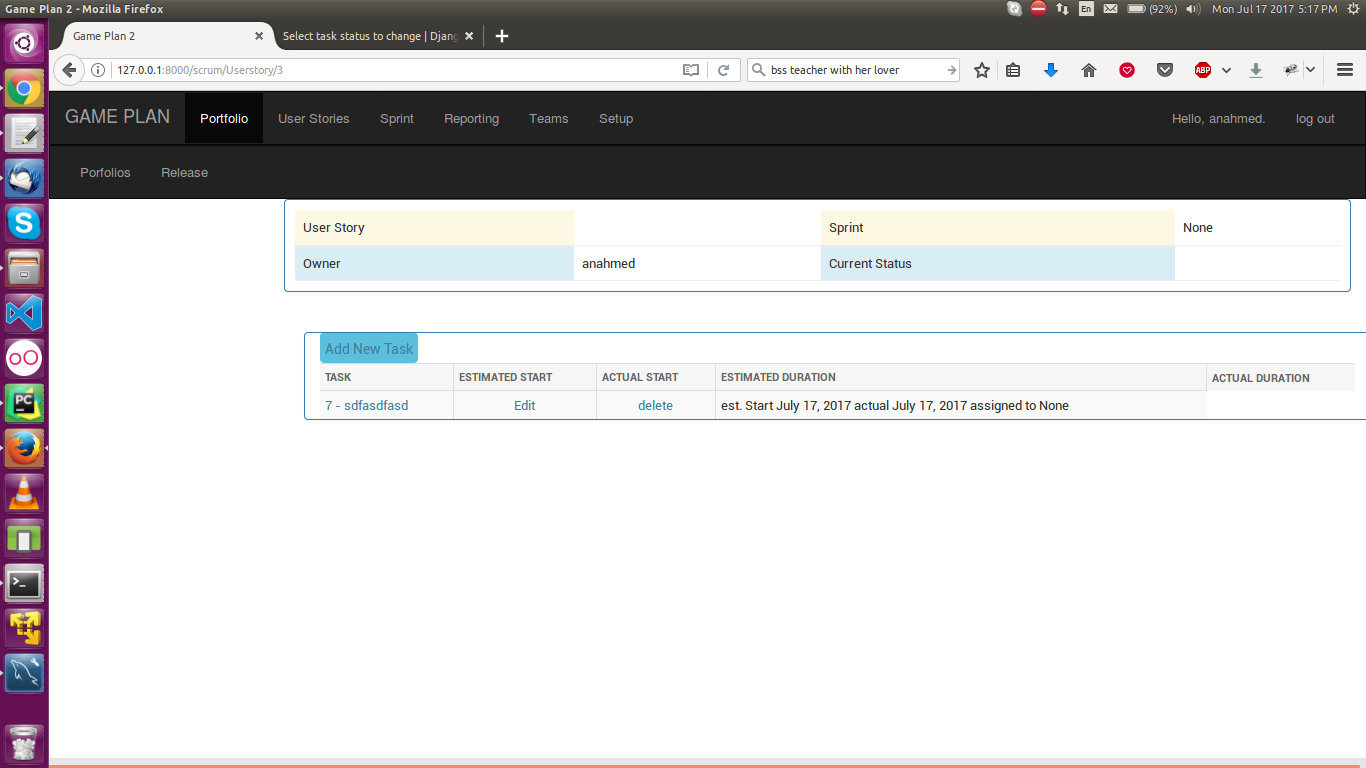
The interactive page of GamePlan2 has two section. One section contain user stories that are not assigned and other sprint of this release.

**Design Details:**

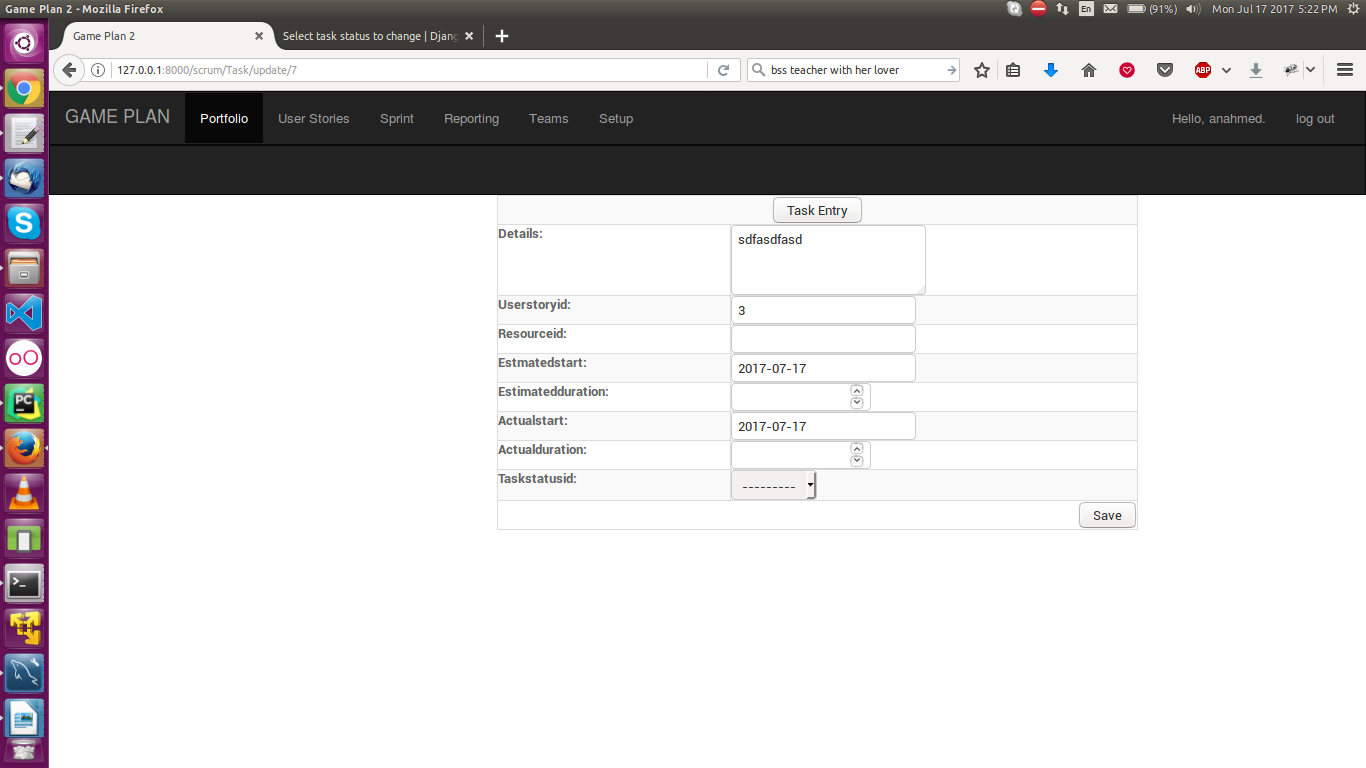
Main url : http://127.0.0.1:8000/scrum

|  |  |  |  |
| --- | --- | --- | --- |
| URL | View | Template | Model |
| **^** **userstory**  (Portfolio details page, showing releases) | UserstoryList | UserstoryList.html | Userstory |
| **^ userstory /(?P<pk>\d+)$**  (Details of selected release, showing user stories and sprints) | UserstoryDetails | UserstoryDetails.html | Userstory |
| **^ userstory /add** | UserstoryCreate | userstory \_form.html | Userstory |
| **^ userstory /update/(?P<pk>\d+)$** | UserstoryUpdate | userstory \_form.html | Userstory |
| **^ userstory /delete/(?P<pk>\d+)$**  Deleting existing release | UserstoryDelete | userstory \_delete.html | Userstory |

# sprint planning: choosing user stories and plan development



# sprint Execution: Peform DEVELOPMENT, test, review and log task against user stories

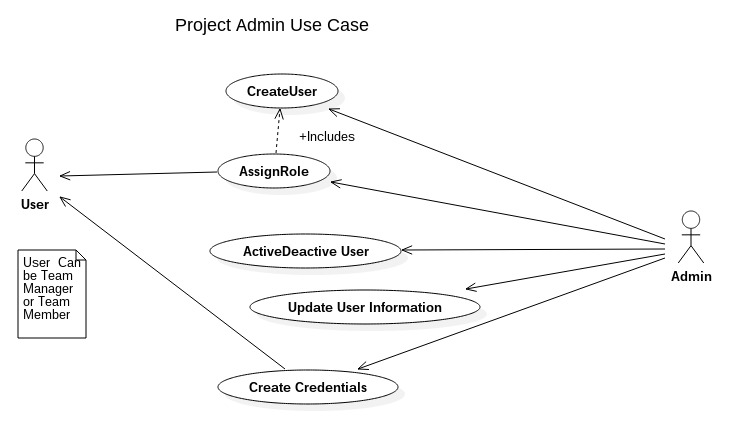


# Core Technolgies

# Key Functions

|  |  |
| --- | --- |
| **Project Dashboard**  Provides progress of in-working projects at a glance over a single screen. It includes Current Status, Total and Pending tasks and assigned team etc. | |
| **Role based Functionalities**  Different users are assigned different roles and could access different functionalities as per their roles. For example only admin roles could update teams and other configuration tasks. | **Project and Team Mapping**  Projects are created and assigned to different teams lead me team manager. |
| **Task Management**  Tasks are assigned to a selected user of project development team. Assigned users could work and update hours to task which will reflects over **project dashboard.** | **Team Management**  Teams are comprised user or resources. A new resource and be registered in the system and assigned different roles as per requirement and hiring contract. |
| **Project reporting and monitoring**  Different screens are available to monitor project progress and adjust project plans according changing environment. | **Project Archiving**  A project can be archived. It will remove from all monitoring screens and dashboard. |

# User Management in gameplan 2



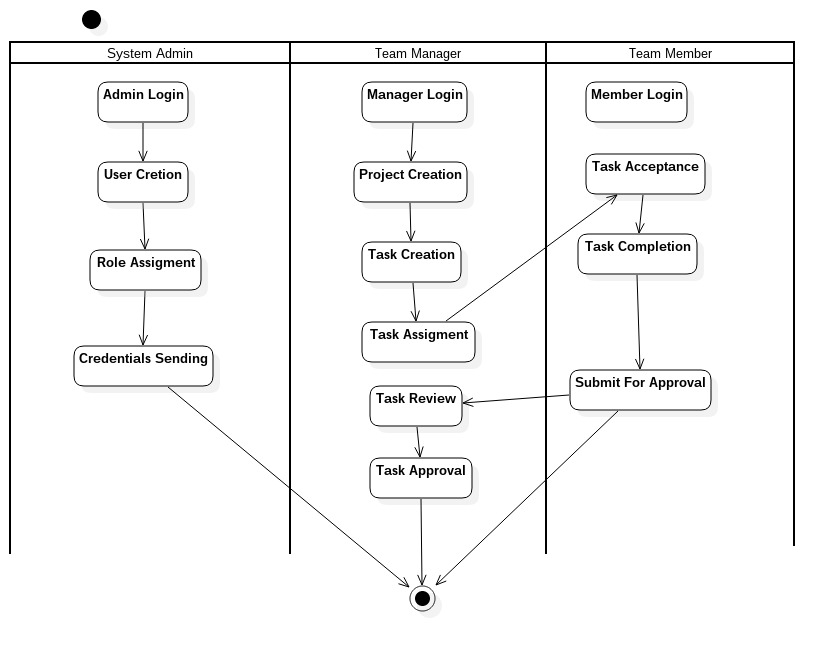


Figure 7: user management

# Entity Relationship Diagram

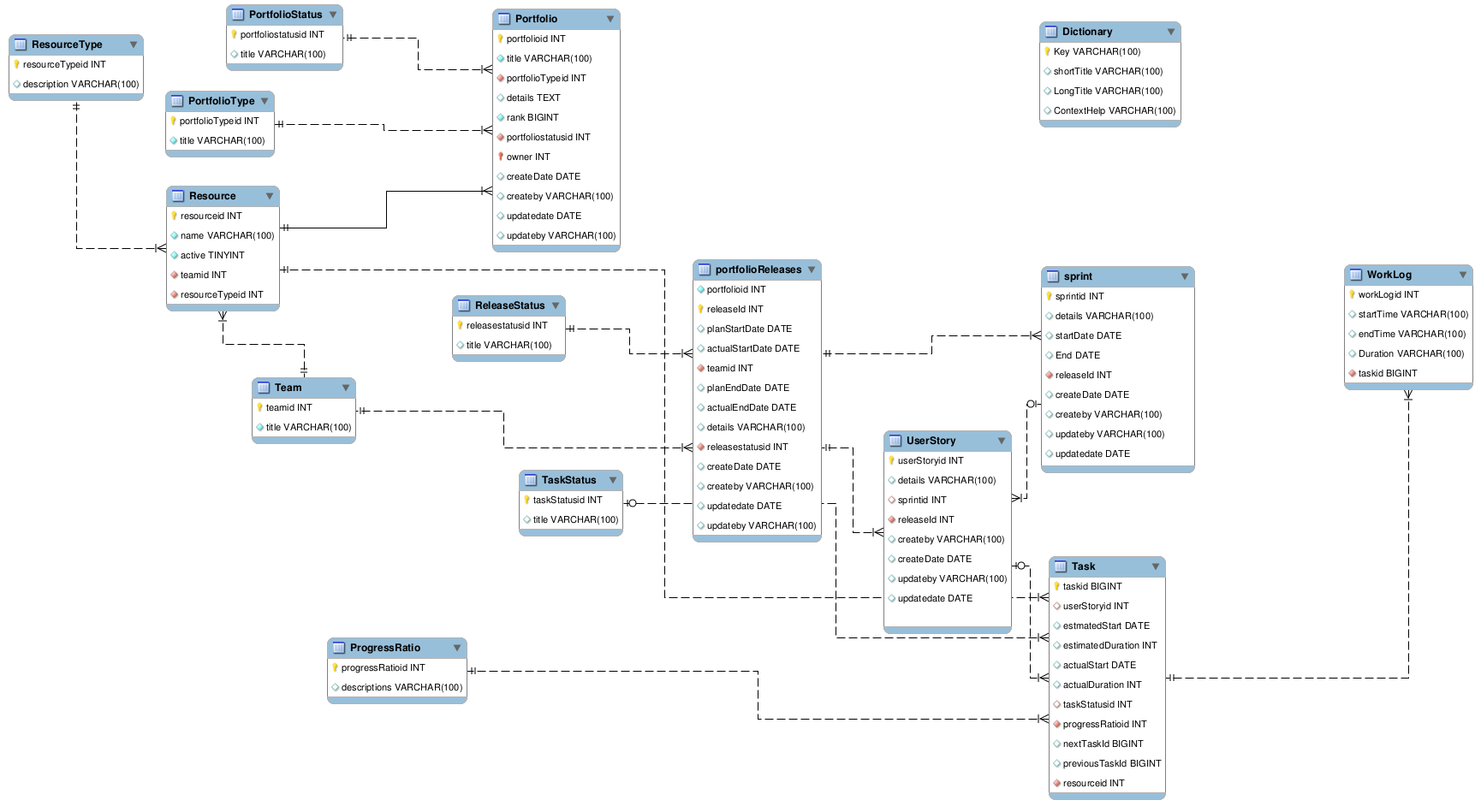


Figure 8: Er diagram