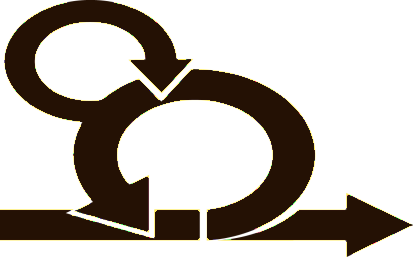
Final Year Project

(Advisor: Sohail Imran)

#### **Game plan 2**



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

Anees Ahmed (58875) - MCS

Table of Contents

[**Table of Figures** **Error! Bookmark not defined.**](#_Toc488327607)

[An Overview of Software Planning with Scrum Framework 4](#_Toc488327608)

[Scrum Flow 5](#_Toc488327609)

[Core Technolgies 7](#_Toc488327610)

[The Django Framework archtecture 8](#_Toc488327611)

[**GAMEPLAN2** DESIGN and archetecture 9](#_Toc488327612)

[The Synopsis of **GAMEPLAN2** 11](#_Toc488327613)

[Portfolio: The Project Dashboard Board 12](#_Toc488327614)

[Releases: working starts here 14](#_Toc488327615)

[User STORIES: Requirement analysis and product backlog 15](#_Toc488327616)

[sprint planning: choosing user stories and plan development 16](#_Toc488327617)

[sprint Execution: Peform DEVELOPMENT, test, review and log task against user stories 17](#_Toc488327618)

[Key Functions 18](#_Toc488327619)

[User Management in **GAMEPLAN2** 19](#_Toc488327620)

[Entity Relationship Diagram 21](#_Toc488327621)

List of Figures

[Figure 1:short incremental development cycle 4](file:///Z:\activelearning\gameplan2\presentation\project%20report.docx#_Toc488327794)

[Figure 2: the Scrum Process flow 5](file:///Z:\activelearning\gameplan2\presentation\project%20report.docx#_Toc488327795)

[Figure 3: A example of app design 9](#_Toc488327796)

[Figure 4: pORTFOLIO RELEASE DESIGN 10](#_Toc488327797)

[Figure 5: Portfolio project dashboard 12](#_Toc488327798)

[Figure 6: Data entry form 12](file:///Z:\activelearning\gameplan2\presentation\project%20report.docx#_Toc488327799)

[Figure 7: Release Form 14](file:///Z:\activelearning\gameplan2\presentation\project%20report.docx#_Toc488327800)

[Figure 8:Releases 15](file:///Z:\activelearning\gameplan2\presentation\project%20report.docx#_Toc488327801)

[Figure 9: user management 20](#_Toc488327802)

[Figure 10: Er diagram 21](file:///Z:\activelearning\gameplan2\presentation\project%20report.docx#_Toc488327803)

# An Overview of Software Planning with Scrum Framework

**GAMEPLAN2** 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 size 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

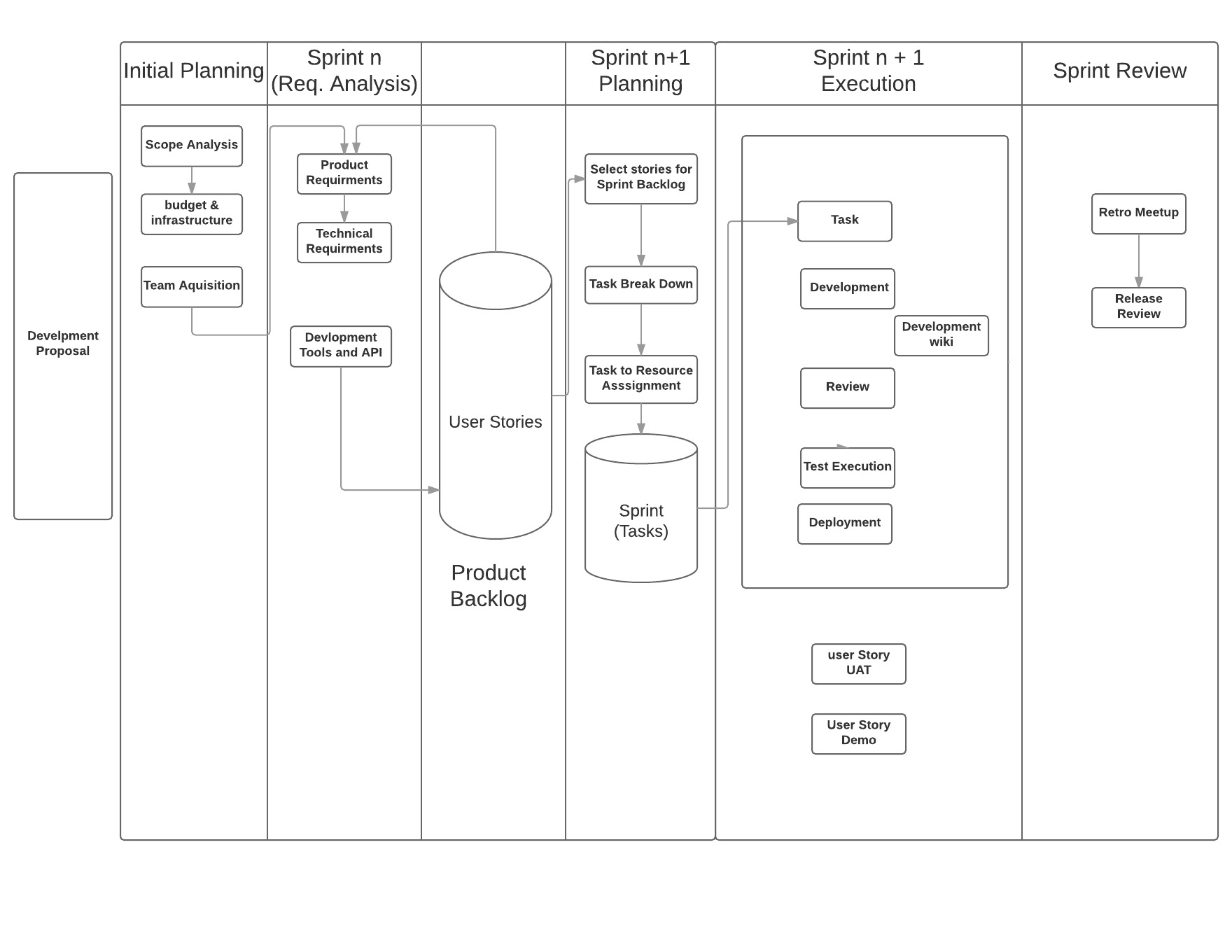
# Scrum Flow

Scrum project plan flows with following phases, from beginning to end. Its requirement analysis, sprint planning and sprint execution are iterative while initial two phase are linear.

|  |  |  |
| --- | --- | --- |
| 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 most attractive feature in scrum framework is its parallel development + Requirement Analysis approach. Unlike other agile methodology, Scrum team does not depends over a requirement to finish. Instead, they choose



# Core Technolgies

# The Django Framework archtecture

**MVT:**

Django offers Model-View-Template Architecture. Views are used as controller and invoked by web-URL request. The fetches data from Models and mixed business logic and filter using built-in classes (generic views and forms) and forward the output to template. The template are usually converted as HTML or JSON codes for browser or native UI libraries.

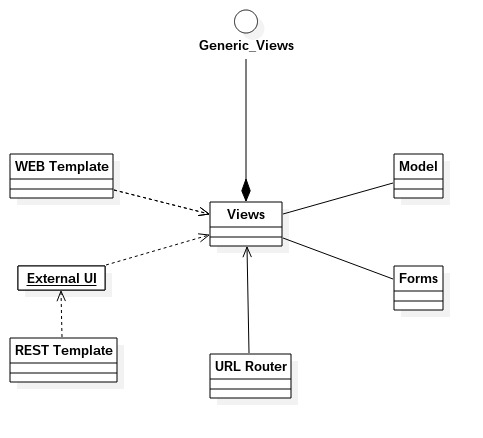
**ORM:**

Django is a python based framework for developing application with built features like Object Relational Mapper (ORM). The ORM provide extended functionality for developing application that could use multiple data store like Oracle, MySQL or even No-SQL. Using ORM we let django implement details and focus our model centric application development.

**Template Engine:**

Django template engine is used to generate WEB UI (HTML, JS) . The engine is capable to work seamlessly with Django ORM without tightly coupled codding. Developer could create codes in related classes and use such codes in the template.

Django Template can also be used to produced JSON output for Mobile and other native client library.



# **GAMEPLAN2** DESIGN and archetecture

The **GAMEPLAN2** application architecture follows Django’s MVT convention. The application uses model first approach, which emphases creating model as app skeleton and continue to developer wrappers like views, routers, forms and template over such models.

Using model first approach also enable to start implementing data store related codes like database schema creation etc. Django’s offers a functionalities that could automatic creates database schema (even for multiple type of data-store like Oracle, MySQL or SQLanywhere).

A model class is used as a core in different template and forms as per UI requirements. Furthermore, it helps us to create views for selecting, filtering, and formatting data with other design needs and UI requirements.

In **GAMEPLAN2**, we take helper interfaces and its generics built-in classes. These helper classes enable creating UI of different style like Dashboard, List, CRUD operations etc.

Following pictures show how our Portfolio Dashboard (main screen) has constructed using Django’s ORM and MVT style. The portfolio board start from view (portfolio list) as main and using model, template and forms create a UI for user interaction.

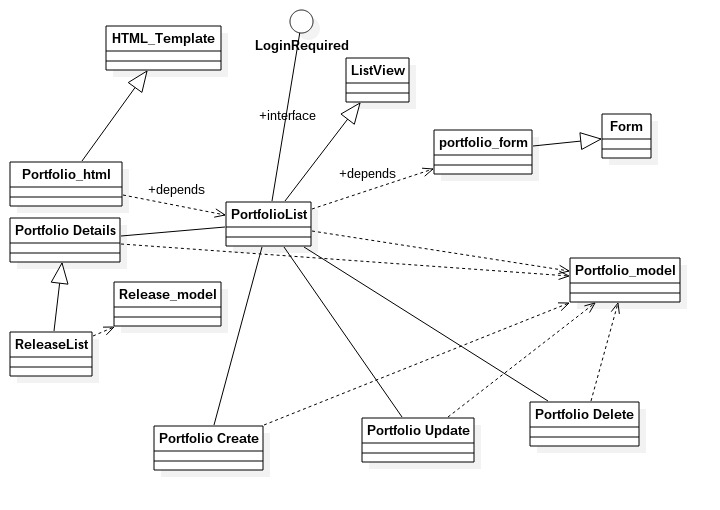


Figure : A example of app design

The following diagram shows another screen of (portfolio releases) that are linked to older screen and extend the UI to next screen. However the convention and architecture is same, proving loose couple classes interaction without compromising flexibility.

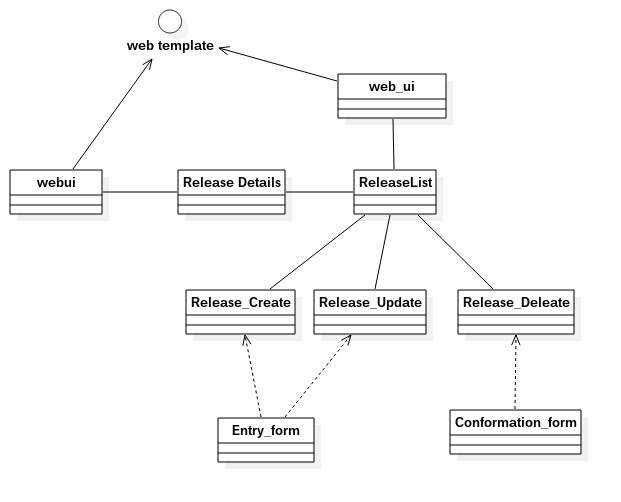


Figure : pORTFOLIO RELEASE DESIGN

# The Synopsis of **GAMEPLAN2**

|  |  |
| --- | --- |
| 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

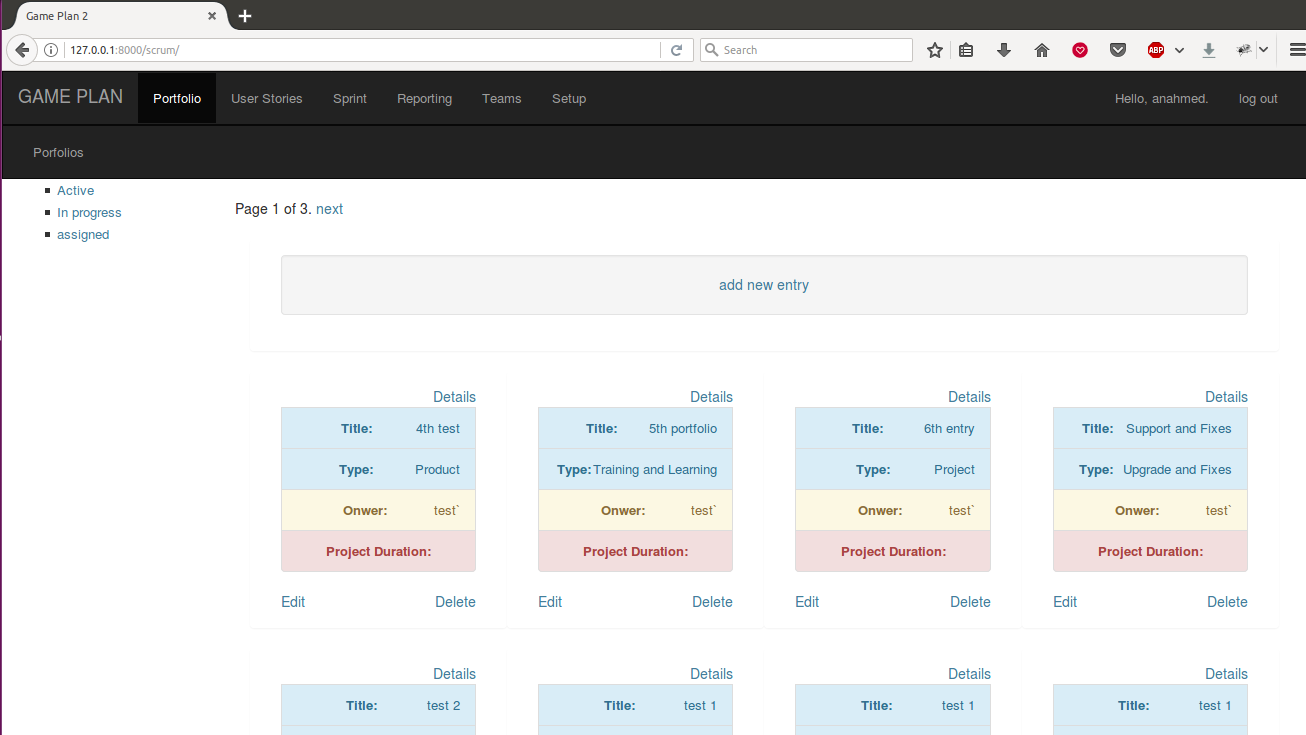


Figure 5: Portfolio project dashboard

Portfolio board or project dashboard is starting point for our **GAMEPLAN2** 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.

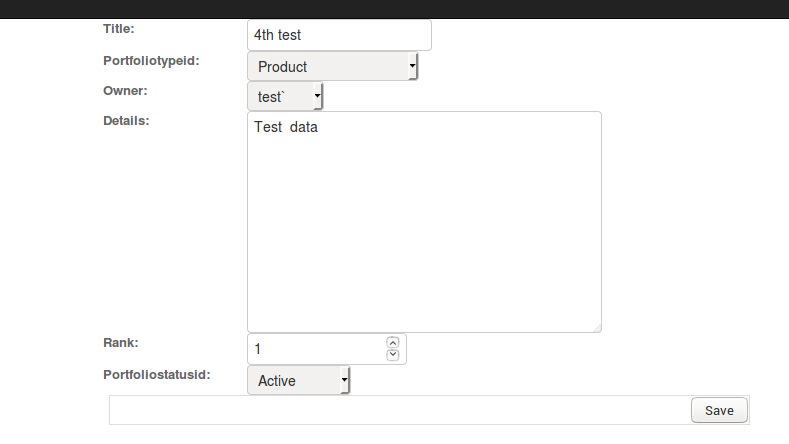


Figure 6: Data entry form

**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

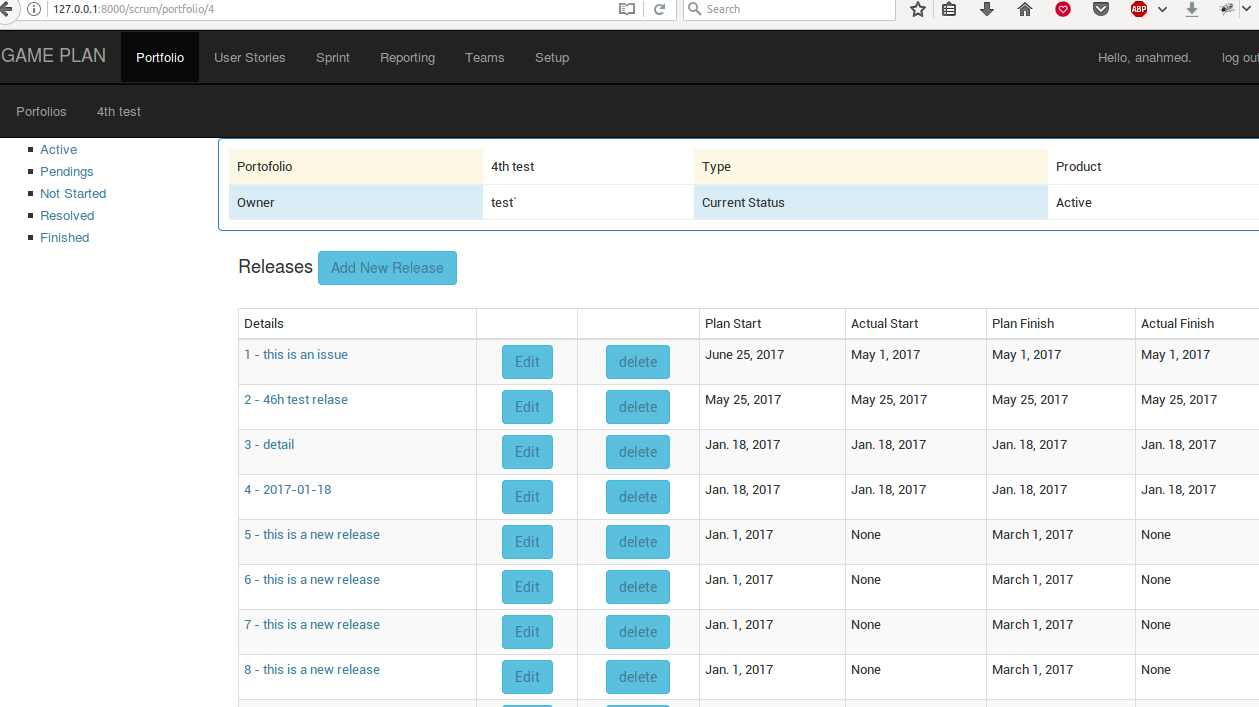


Figure 7: Release Form

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 8:Releases

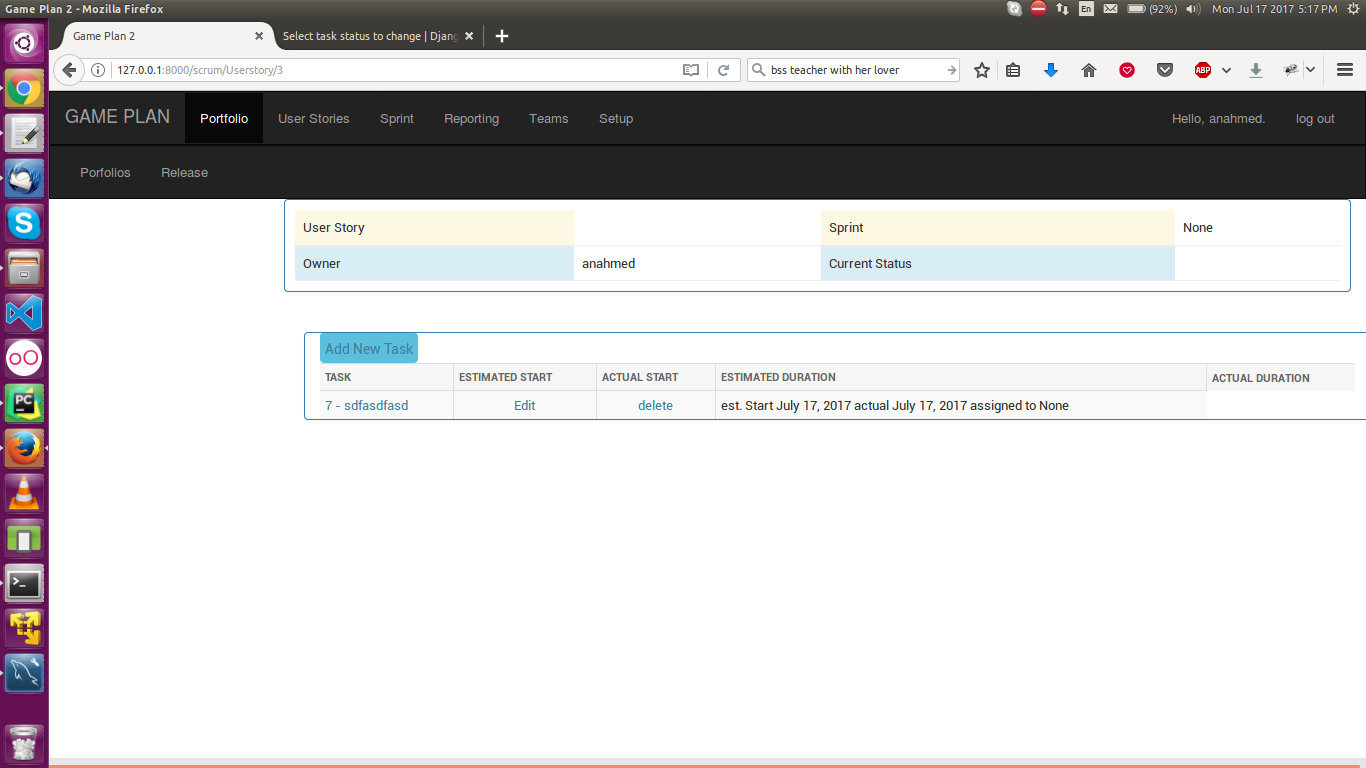
The interactive page of GAMEPLAN2 has two section. One section contain user stories that are not assigned and other sprint of this release.

**Design Details:**

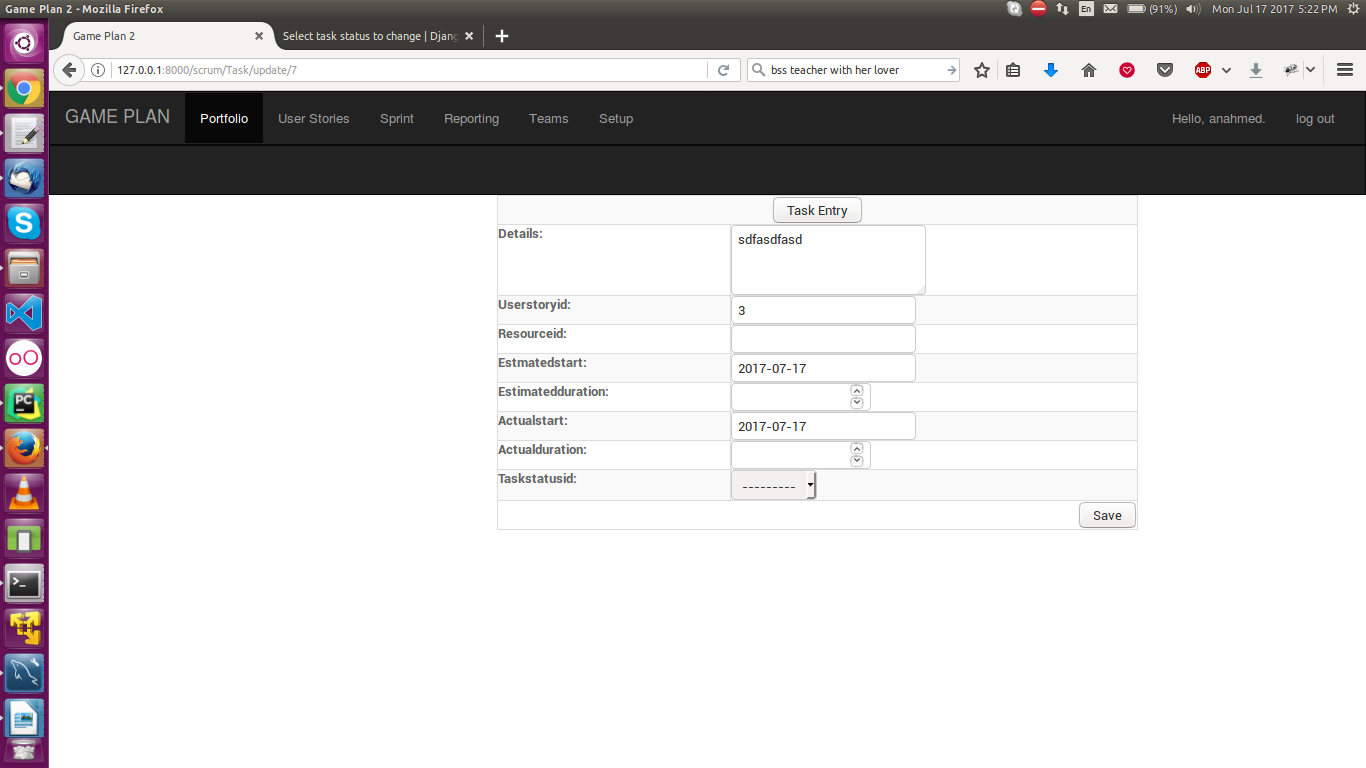
Main url : http://127.0.0.1:8000/scrum

|  |  |  |  |
| --- | --- | --- | --- |
| 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



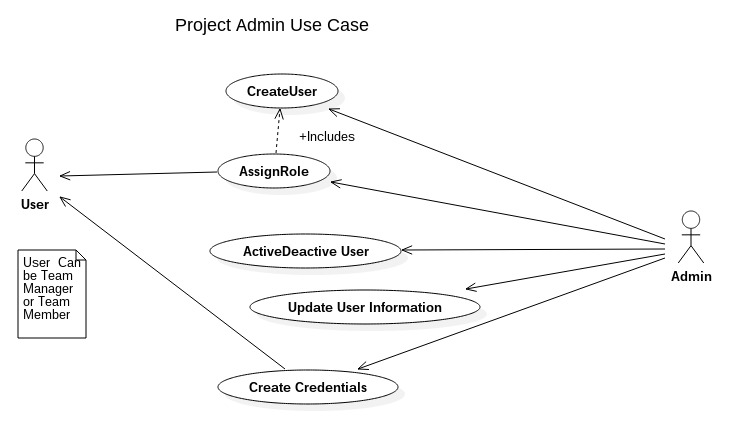
# sprint Execution: Peform DEVELOPMENT, test, review and log task against user stories



# 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 **GAMEPLAN2**



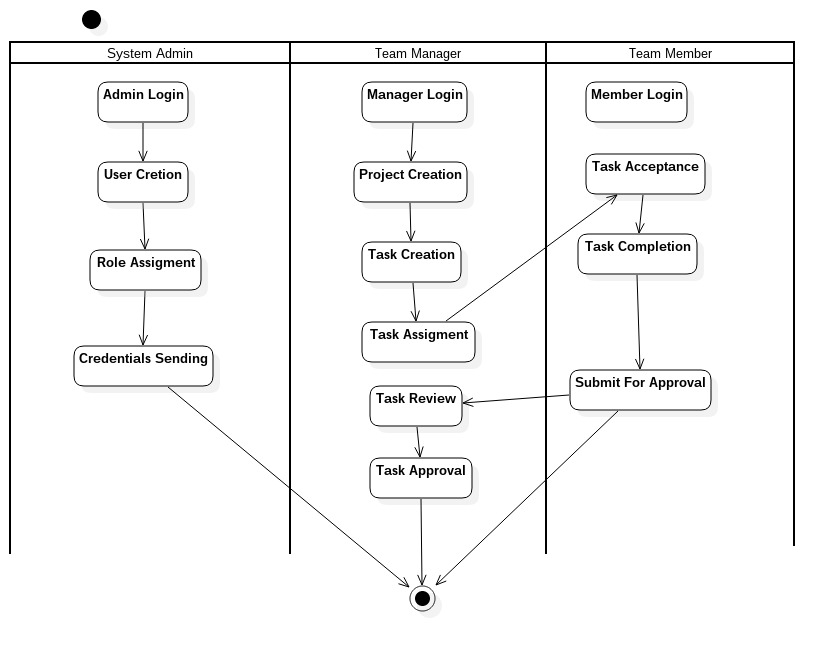


Figure 9: user management

# Entity Relationship Diagram

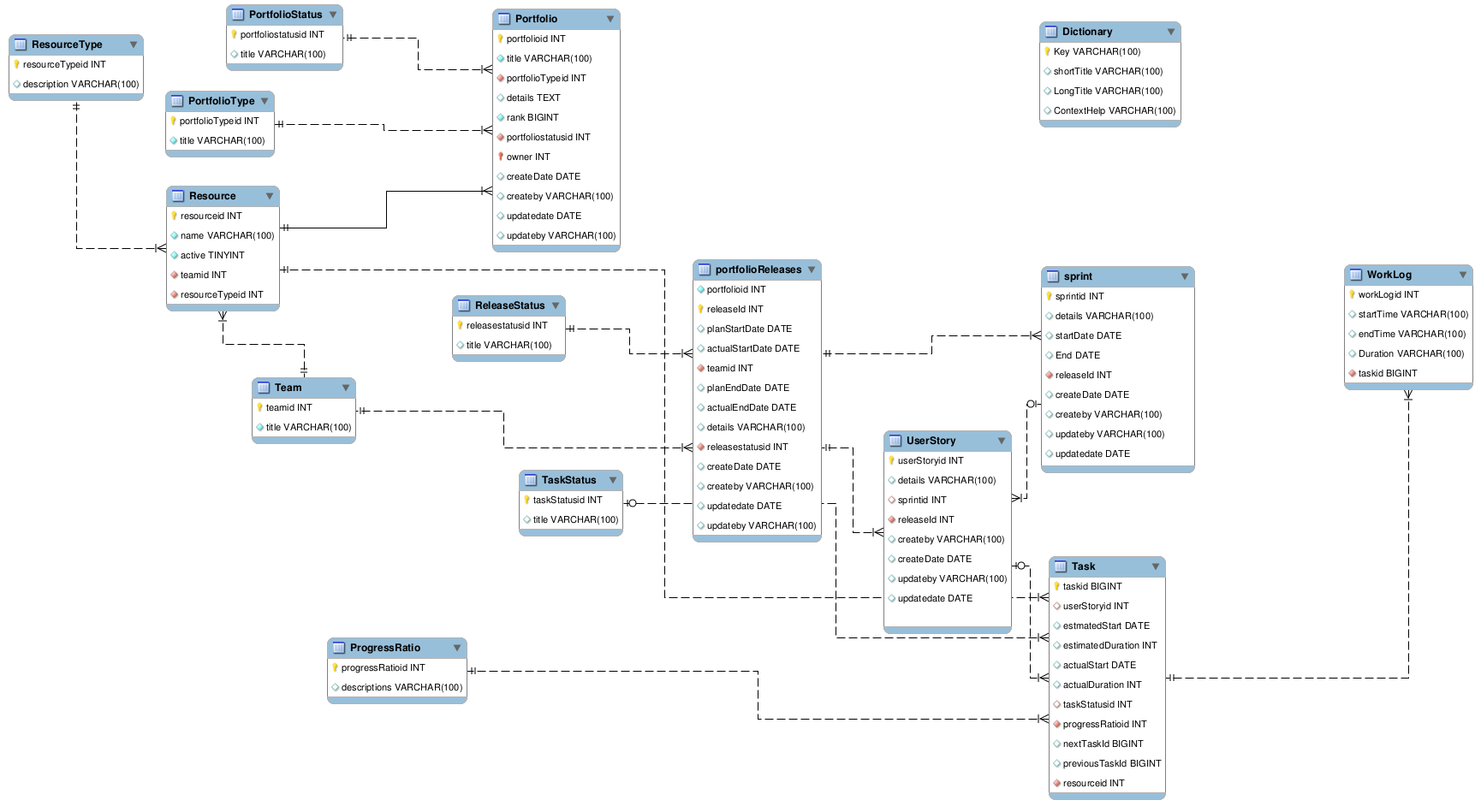


Figure 10: Er diagram