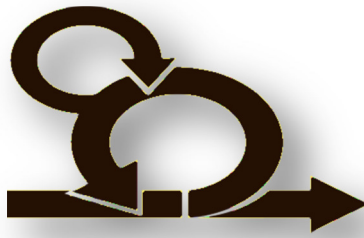


Final Year Project  
(Advisor: Sohail Imran)

---

# GAME PLAN 2

---



## SOFTWARE PLANNING WITH *SCRUM* FRAMEWORK

Anees Ahmed (58875) - MCS

## TABLE OF CONTENTS

<b>Table of Figures.....</b>	<b>Error! Bookmark not defined.</b>
An Overview of Software Planning with Scrum Framework.....	4
Scrum Flow.....	5
Core Technolgies.....	7
The Django Framework architecture.....	8
<b>GAMEPLAN2</b> DESIGN and archetecture.....	9
The Synopsis of <b>GAMEPLAN2</b> .....	11
Portfolio: The Project Dashboard Board.....	12
Releases: working starts here .....	14
User STORIES: Requirement analysis and product backlog .....	15
sprint planning: choosing user stories and plan development .....	16
sprint Execution: Peform DEVELOPMENT, test, review and log task against user stories.....	17
Key Functions.....	18
User Management in <b>GAMEPLAN2</b> .....	19
Entity Relationship Diagram.....	21

---

## LIST OF FIGURES

---

Figure 1: short incremental development cycle	4
Figure 2: the Scrum Process flow	5
Figure 3: A example of app design	9
Figure 4: pORTFOLIO RELEASE DESIGN	10
Figure 5: Portfolio project dashboard	12
Figure 6: Data entry form	12
Figure 7: Release Form	14
Figure 8: Releases	15
Figure 9: user management	20
Figure 10: Er diagram	21

## 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.

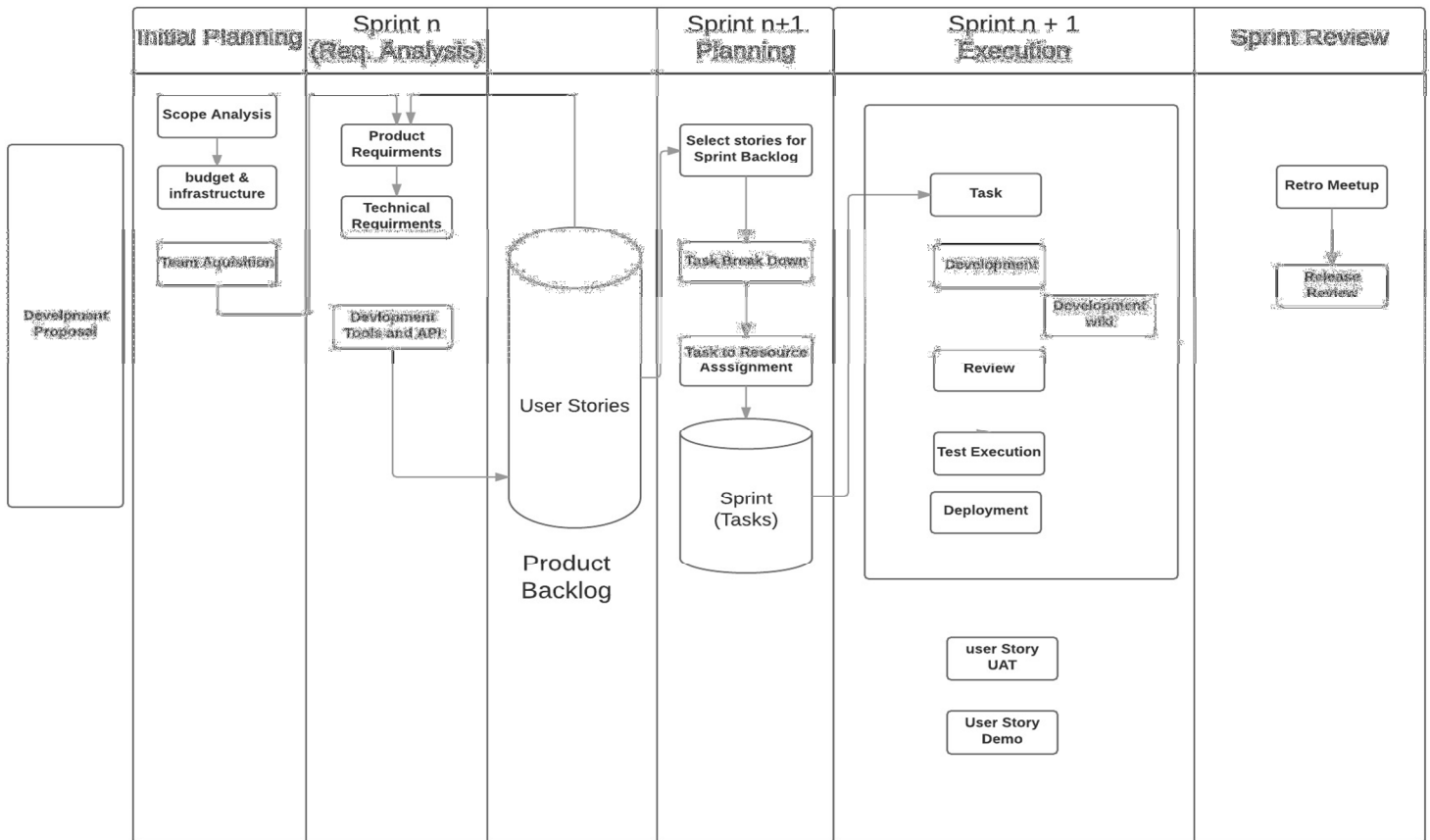


## SCRUM FLOW

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	Scrum 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.

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 TECHNOLOGIES

---

Web UI:  
(HTML, Bootstrap, JQuery)

Sever Technolgy:  
Django , Python

Web Server:  
Apache

database:  
MySQL

## 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.

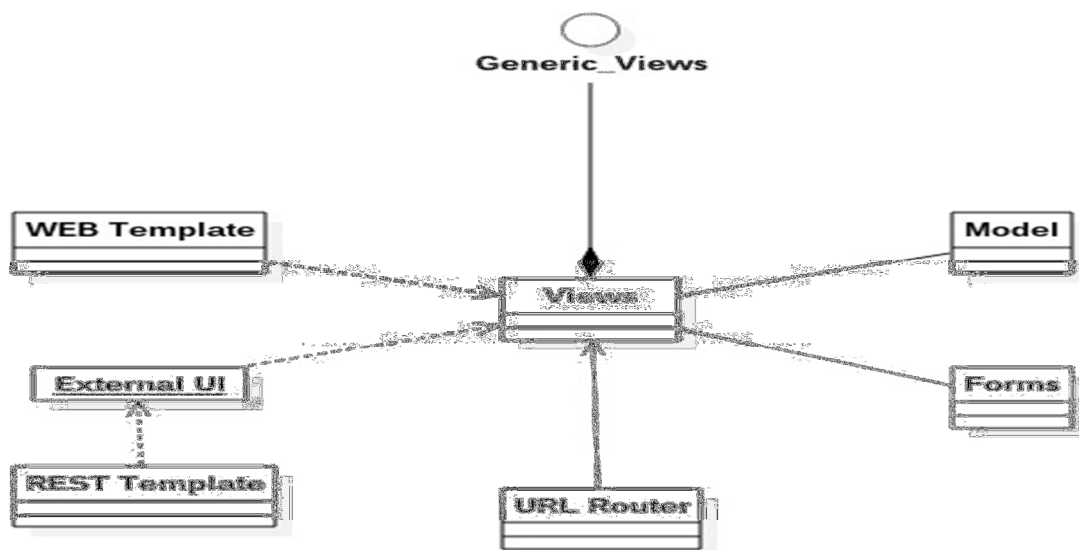
### ORM:

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 coding. 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 emphasizes 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.

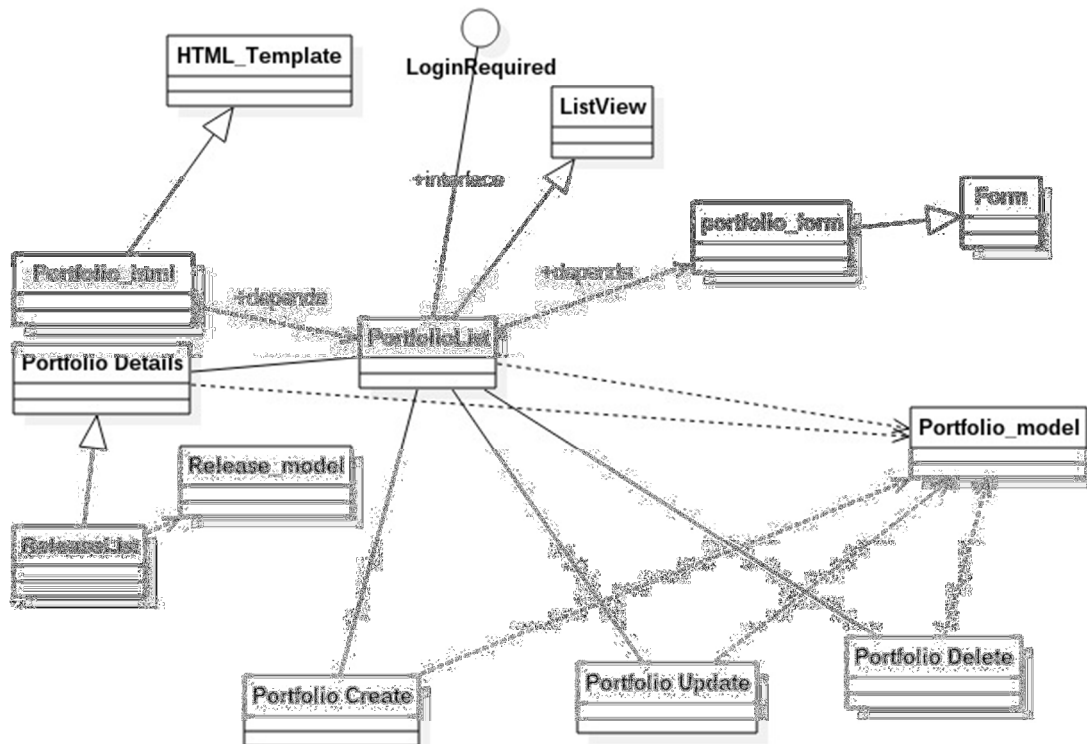


FIGURE 3: A EXAMPLE OF APP DESIGN

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.

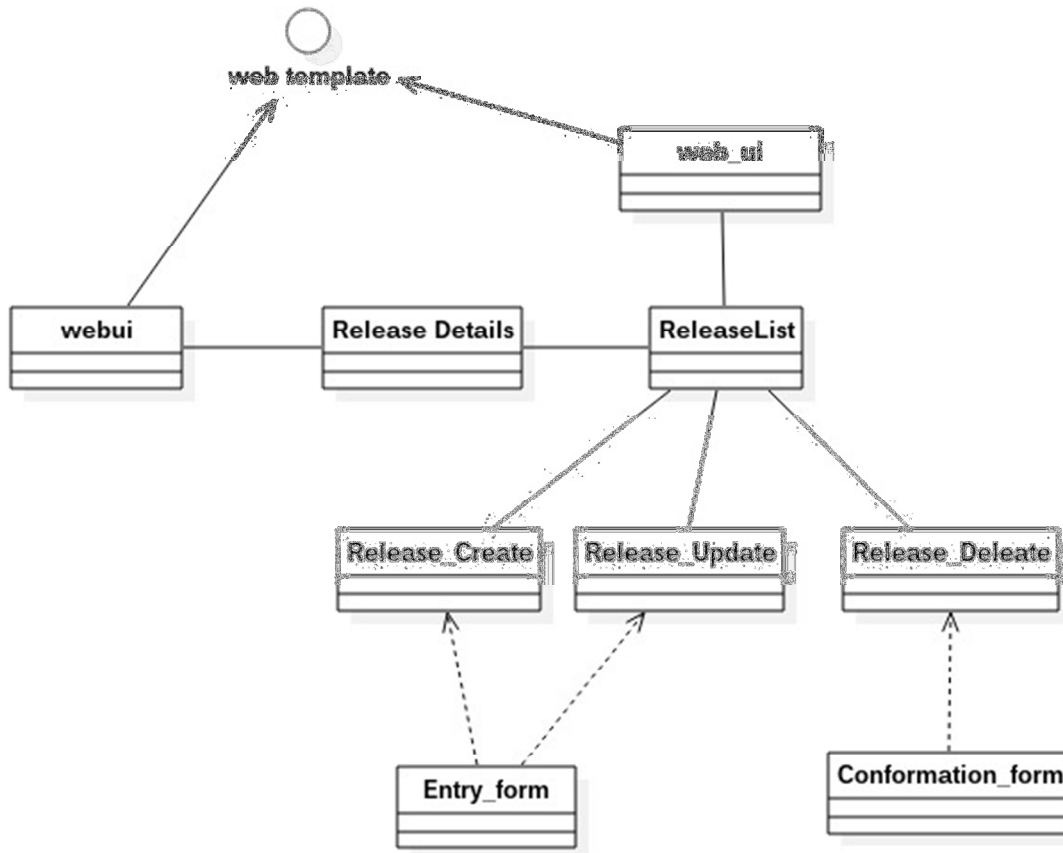


FIGURE 4: PORTFOLIO RELEASE DESIGN

## 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

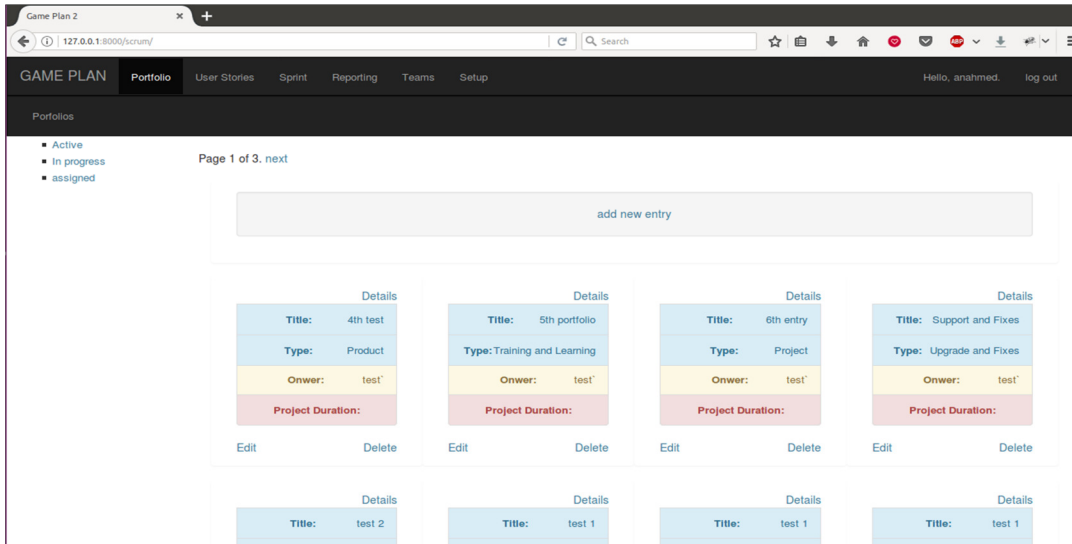


FIGURE 5: PORTFOLIO PROJECT DASHBOARD

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.

Title:

4th test

PortfolioTypeId:

Product

Owner:

test'

Details:

Test data

Rank:

1

PortfolioStatusId:

Active

Save

FIGURE 6: DATA ENTRY FORM

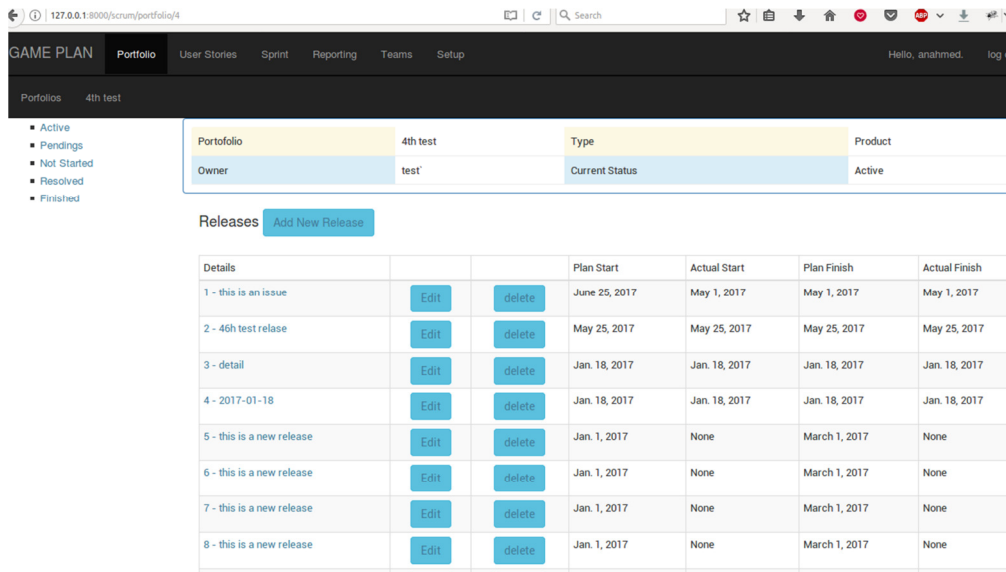
### Design Details:

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
<b>^portfolio</b> (Portfolio Dashboard)	PortfolioList	PortfolioList.html	Portfolio
<b>^portfolio/(?P&lt;pk&gt;\d+)\$</b> (moves to details pages and shows project details and its releases)	PortfolioDetails	PortfolioDetails.html	Portfolio
<b>^portfolio/add</b>	<b>PortfolioCreate</b>	portfolio_form.html	Portfolio
<b>^portfolio/update/(?P&lt;pk&gt;\d+)\$</b>	<b>PortfolioUpdate</b>	portfolio_form.html	Portfolio
<b>^portfolio/delete/(?P&lt;pk&gt;\d+)\$</b>	<b>PortfolioDelete</b>	conform_delete.html	Portfolio

## RELEASES: WORKING STARTS HERE



The screenshot shows the 'Portfolio' page in the GAMEPLAN2 application. The 'Releases' section is active, displaying a table of releases. The table has columns for Details, Edit, delete, Plan Start, Actual Start, Plan Finish, and Actual Finish. The releases are listed as follows:

Details	Edit	delete	Plan Start	Actual Start	Plan Finish	Actual Finish
1 - this is an issue	Edit	delete	June 25, 2017	May 1, 2017	May 1, 2017	May 1, 2017
2 - 46h test release	Edit	delete	May 25, 2017	May 25, 2017	May 25, 2017	May 25, 2017
3 - detail	Edit	delete	Jan. 18, 2017	Jan. 18, 2017	Jan. 18, 2017	Jan. 18, 2017
4 - 2017-01-18	Edit	delete	Jan. 18, 2017	Jan. 18, 2017	Jan. 18, 2017	Jan. 18, 2017
5 - this is a new release	Edit	delete	Jan. 1, 2017	None	March 1, 2017	None
6 - this is a new release	Edit	delete	Jan. 1, 2017	None	March 1, 2017	None
7 - this is a new release	Edit	delete	Jan. 1, 2017	None	March 1, 2017	None
8 - this is a new release	Edit	delete	Jan. 1, 2017	None	March 1, 2017	None

FIGURE 7: RELEASE FORM

In the detail page of portfolio, 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 methodology (incremental, iterative), more release of a single portfolio are common.

### Design Details:

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

URL	View	Template	Model
<b>^release</b> (Portfolio details page, showing releases)	ReleaseList	ReleaseList.html	Release
<b>^release/(?P&lt;pk&gt;\d+)\$</b> (Details of selected release, showing user stories and sprints)	ReleaseDetails	ReleaseDetails.html	Release
<b>^release/add</b>	ReleaseCreate	release_form.html	Release
<b>^portfolio/update/(?P&lt;pk&gt;\d+)\$</b>	ReleaseUpdate	release_form.html	Release
<b>^release/delete/(?P&lt;pk&gt;\d+)\$</b> 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.

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/scrums>

URL	View	Template	Model
<b>^ userstory</b> (Portfolio details page, showing releases)	UserstoryList	UserstoryList.html	Userstory

FIGURE 8:RELEASES

<b>^ userstory /(P&lt;pk&gt;d+)\$</b> (Details of selected release, showing user stories and sprints)	UserstoryDetails	UserstoryDetails.html	Userstory
<b>^ userstory /add</b>	UserstoryCreate	userstory _form.html	Userstory
<b>^ userstory /update/(P&lt;pk&gt;d+)\$</b>	UserstoryUpdate	userstory _form.html	Userstory
<b>^ userstory /delete/(P&lt;pk&gt;d+)\$</b> Deleting existing release	UserstoryDelete	userstory _delete.html	Userstory

## SPRINT PLANNING: CHOOSING USER STORIES AND PLAN DEVELOPMENT

Game Plan 2 - Mozilla Firefox

Game Plan 2 x Select task status to change | Djan: x +

127.0.0.1:8000/scrum/Userstory/3

bsa teacher with her lover

GAME PLAN Portfolio User Stories Sprint Reporting Teams Setup Hello, anahmed. log out

Portfolios Release

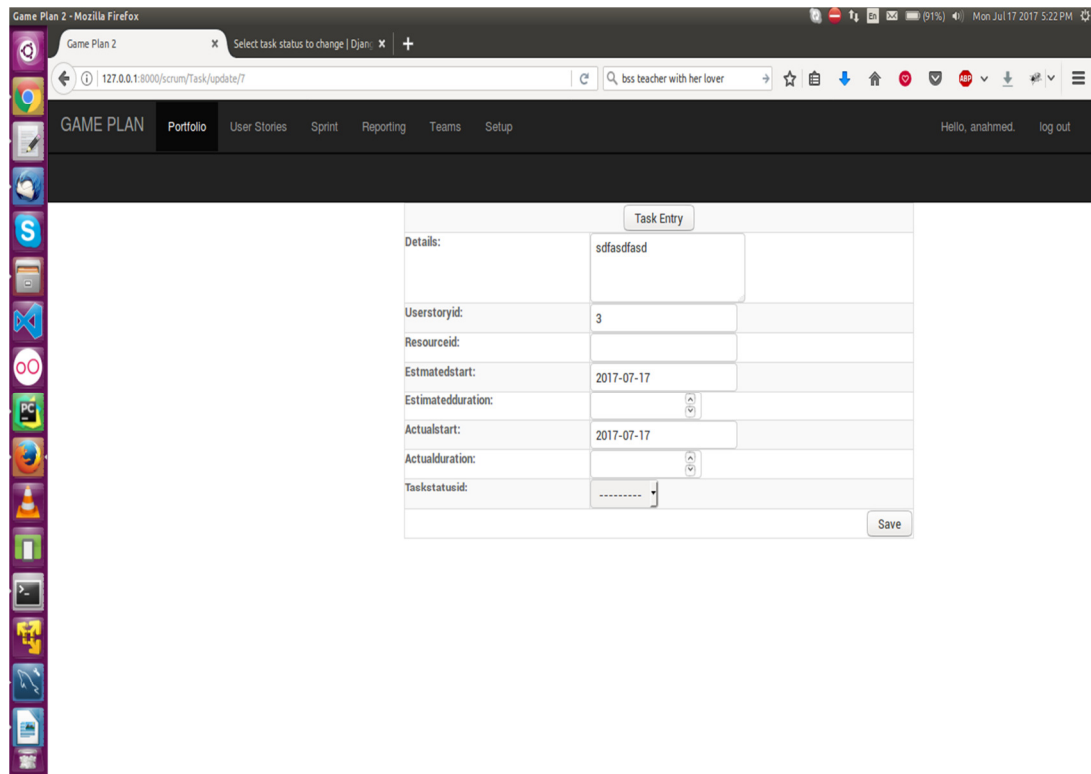
User Story	Sprint
Owner: anahmed	Current Status: None

Add New Task

TASK	ESTIMATED START	ACTUAL START	ESTIMATED DURATION	ACTUAL DURATION
7 - sdfasdfsad	Edit	delete	est. Start July 17, 2017 actual July 17, 2017 assigned to None	



## SPRINT EXECUTION: PEFORM DEVELOPMENT, TEST, REVIEW AND LOG TASK AGAINST USER STORIES



The screenshot shows a web browser window titled "Game Plan 2 - Mozilla Firefox". The address bar shows the URL "127.0.0.1:8000/scrum/Task/update/7". The page has a navigation bar with tabs: "GAME PLAN", "Portfolio", "User Stories", "Sprint", "Reporting", "Teams", and "Setup". The "Portfolio" tab is active. The main content area displays a "Task Entry" form with the following fields:

Details:	sdfasdfsad
Userstoryid:	3
Resourceid:	
Estimatedstart:	2017-07-17
Estimatedduration:	
Actualstart:	2017-07-17
Actualduration:	
Taskstatusid:	-----

A "Save" button is located at the bottom right of the form.

## 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

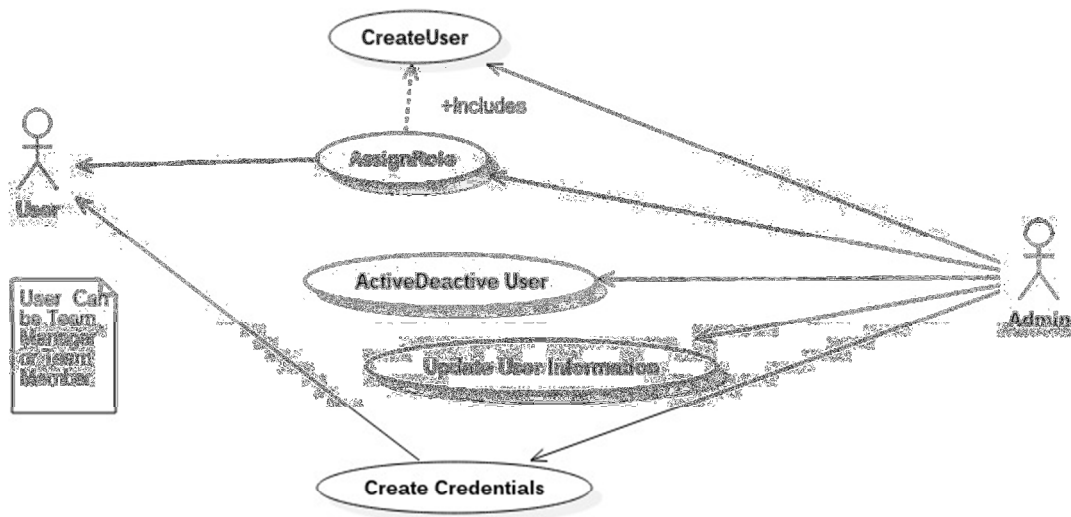
Different screens are available to monitor project progress and adjust project plans according changing environment.

### Project Archiving

A project can be archived. It will remove from all monitoring screens and dashboard.

## USER MANAGEMENT IN GAMEPLAN2

### Project Admin Use Case



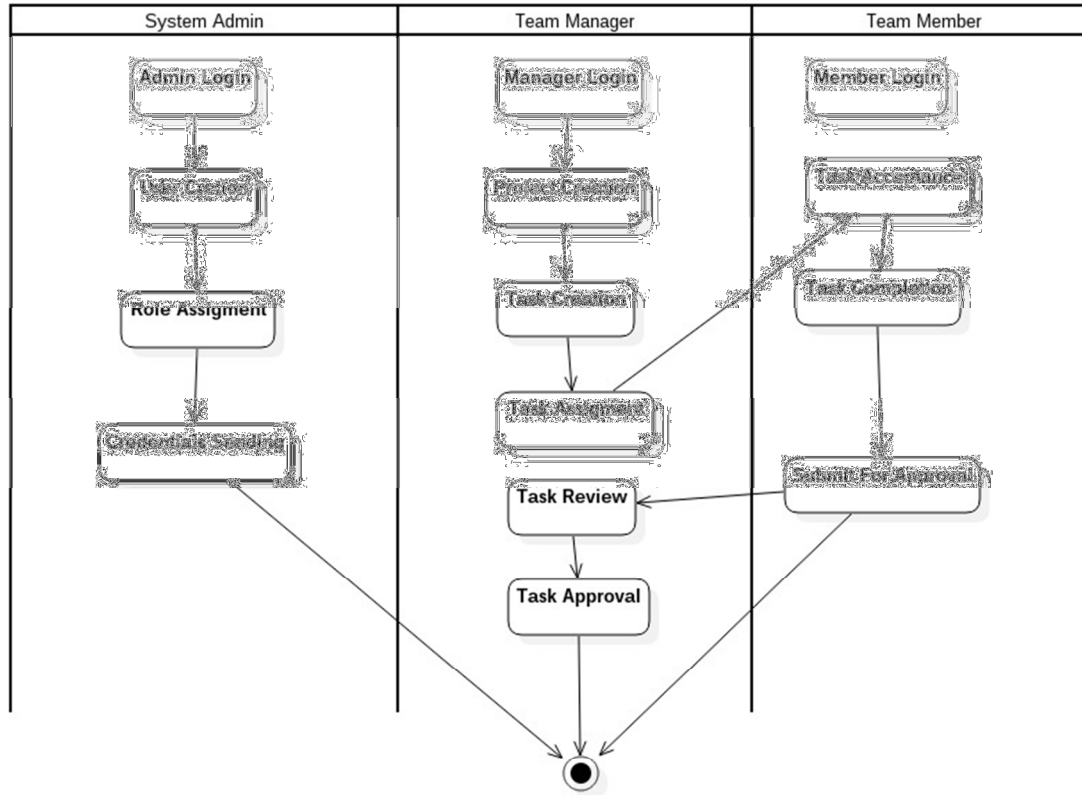


FIGURE 9: USER MANAGEMENT

