Software Engineering Project Report

A Sample Document for Ozil-Case Tool

Prepared by
Vinit Kumar
Jesus Solorzano
Omaid Khan
Philip Variciuc
for use in CS 440
at the
University of Illinois Chicago

April 2016

Table of Contents

List of Tables

Date	Participants	Activities	Notes
04/30/2016	Omaid Khan	Section 1,2	Initial Draft
04/30/2016	Filip Variciuc	Section 3, 4,	Initial Draft
04/30/2016	Jesus Solorzano	Section 5,6	Initial Draft
04/30/2016	Vinit Kumar	Section 6,7,8	Initial Draft, Proof Reading

I Project Description

1 Project Overview

The Ozil is another product in the long list of "A CASE Tool to Aid Agile Software Development" type of product. It is mainly targeted for the audience who want to develop their products/or provide IT service and follow agile methodology for their development procedure.

The user can log in to the product as admin and have full access of the project or can log in as one of the team member and have limited accessibility of the project like creating user stories, creating tasks for those stories, assigning/taking up those tasks, updating the task status to different states, logging in defects/assigning the defect to the respective resource, triaging the defect.

The user can also log into the product as test manager and check for the reports such as burndown and burnup charts.

The target audience are people from various organizations be it 2 member team, small startup, to big established firm and for the various users in the organization who may/may not be technically qualified to use our product. One of the major goal/challenge is to make the product as much visually attractive without going totally overboard and to provide the basic and advanced functionality like showing the result to the user as well as giving him the option to host the product on either his network or our network for a small charge.

A brief description of the product to be produced, before getting into details.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

The audience that we are targeting are people from various organizations be it 2 member team, small startup, to big established firm and for the various users in the organization who may/may not be technically qualified to use our product.

We know about the constraints of the various organizations be it physical or budget resources and provide our plans accordingly. For the bigger client, we provide teams working exclusively for them

2b Goals of the Project

The goal is to provide our customers a product that is as much visually attractive as much it is functional as well as giving him the option to host the product on either his network/server or our network/server for appropriate charge. The user should feel completely at home using our product and we should as well be providing good support for our product.

2c Measurement

The activities, such as creating user stories, creating tasks for those stories, assigning/taking up those tasks, updating the task status to different states, logging in defects/assigning the defect to the respective resource, triaging the defect, for the particular sprint is documented and stored in the database against each user for that that particular task in the form of quantifiable number.

The burndown/burnup report is generated at the end of the sprint activities and is a further tool for visualizing or crunching the numbers in the context of project.

3 The Scope of the Work

3a The Current Situation

Currently, there are good many software/product in the market that comes within the definition of "A CASE Tool to Aid Agile Software Development" but most of them are expensive high end product and cheaper version of them don't necessarily offers enough features to support the development task. Our Product deals with providing the same features and some additional features like font, transition smoothing, customized graphs for a basic price and also provide the customer with the option of hosting the data on our server or at their server at a minimum cost.

Our main focus is to provide the customers with a very friendly easily identifiable tools with lots of video tutorials and tool tips as hints, so that he can easily get acclimatized to our product and along with good UI and decent amount of animation, good variety of graphical representation of the final burndown/burnup charts.

We also will be releasing the product on mobile/handheld/tablet/phablet devices in the future release.

3b The Context of the Work

Not Applicable

3c Work Partitioning

Event Name	Input and Output	Summary
1. User creates new login	Save the user profile(in)	Save the user profile in the database.
2. User logs in the system with the newly created	Login activity(in)	Check whether the user profile is correct
3. User creates user story	Save the story(in)	Save the story in the database
4. User creates task for the user stories	Saves the task(in)	Save the task in the database and map it to the story for which it is created
5. User assigns the tasks to different resources	Save the tasks to the resources(in)	Save the setting in the database and map the resources to which the user has mapped the tasks
6. User moves the task cards across various states.	Change the status of the cards appropriately(out)	Appropriately update the status of the task in the database and display the same result in the front end
7. User logs defect as a part of task	Store the defect(in)	Save the defect and automatically assign it a unique ID
8. User triages the defect.	Change the status of the defect(in)	Change the status of the defect in the database as per the user changes
9. User logs in the system as manager and checks for the burndown/burnup charts	Display the charts to the user(out)	Fetch the numbers and call for the functions according to the graphs that the user want to see

3d Competing Products

Competing products include all the different tools that aid agile software development However, what we can provide that is different from the competition is better UI, training, fonts, graphs, support for our product.

4 The Scope of the Product

The purpose of this section is to define what is included in the product to be created, and what is not. It is also important to identify what entities external to the product interact directly with the product, and in what manner. This then defines the boundary of the product and the interface it has with external entities (which may be humans, hardware, other sofware systems, etc.)

4a Scenario Diagram(s)

Not Applicable

4b Product Scenario List

(1) User login/register into the system

(2) User creates user story

(3) User creates task for the user stories

(4) User assigns the task to the different resources in the team

(5) User moves the task across various stages.

(6) User creates defect for the current sprint

(7) User triages the defect

(8) User logs in as manager and checks for the various report

4c Individual Product Scenarios

1) New user will register and returning user will login

2) User creates user story: The user converts the technical jargon of the customer to functional requirements in the form user story, a set of scenarios which notifies the developer or the tester who are the actors and external system, how they interact and what to expect from the interaction

3) User creates task for the user stories: After the user story has been accepted by the users, they are then divided in the form of tasks which themselves may contain a list pf subtasks to be performed for the task to be completed.

4) User assigns the task to the different resources in the team: After creation of task, the user assigns the tasks amongst the team member and chooses whichever person is the most suitable fit for the task.

5) User moves the task across various stages: The user then changes the state of the task from ToDo to In Progress to Done

6) User creates defect for the current sprint: User logs in defect/raises issue that he finds in the requirement document or while testing the application and sets the severity, priorty of the defect and assigns the defect to the appropriate developer.

7) User triages the defect: The user after logging the defect then triages the defect and changes the status of the defect accordingly.

8) User logs in as manager and checks for the various report: User has the option to check and generate report for the efforts that been put in by the team for the current sprint and he can generate the burndown/burnup chart.

5 Stakeholders

5a The Client

The client for Ozil will be the set of users who want to develop their projects in agile methodology.

These user's group size can vary, ranging from a group of 2 people, to a small start up firm to a big established organization.

5b The Customer

Activities of a particular sprint like creating task, assigning the task and dividing the work amongst the various resources in the team, updating the status of the task, opening/closing defect/s and generating the final burndown chart is available to the customer. He can use our software as an interface to store/view/update his results and either host the database at his end or chose our server.

In any case, the customer will be one who uses our software for day to day task tracking activity for a particular sprint

At the end of the sprint, during retrospective meeting or at the beginning of the next sprint, during sprint planning meeting, he can use our software to check how well the team performed, how many "complex" tasks was completed in that sprint, how many new tasks should the team should commit in the next meeting.

5c Hands-On Users of the Product

The hands-on users can be broadly divided into 6 categories. The categories are as follows:

User Name: business analyst

User Role: The first category is for the business analyst to sit with the client, and convert the technical jargon into functional requirement format. He creates the user story in the Ozil software and after discussion with both the Test and Development Manager passes on the requirement to the Testing team.

Subject matter experience: They are generally subject matter experts of their field and have knowledge to convert the technical jargon to simple user stories.

Technological experience: They have fair amount of experience in both their field/domain (banking, healthcare, etc.) as well as good amount of experience in software field.

Other user characteristics:

Physical abilities/disabilities: Does not matter

Intellectual abilities/disabilities: Should be expert, so, ideally no intellectual disability

Attitude toward job: Very enthusiastic. The job is quite tough and requires a good attitude.

Attitude toward technology: Moderately experienced. They should know the basic stuff.

Education: Subject Matter Experts. Therefore a good level of education or experience in the domain

Linguistic skills: Very much required.

User Name: Software Test Engineers

User Role: The second category consists of the group of software test engineers who are working on the task. They can be either manual testers who create the test cases and update the status of their task in the Ozil product or automation testers who begin their task of scripting those automatable test cases and updating their status in the Ozil product/software respectively.

Subject matter experience: They should know the in and out of the product along with fair amount of scripting/coding ability and tools for performing the tasks.

Technological experience: They should have high amount of experience in finding the loop hole in the system that is been tested along with fair amount of scripting/coding ability and tools for performing the tasks.

Other user characteristics:

Physical abilities/disabilities: Does not matter

Attitude toward job: Very enthusiastic. The job is quite tough and requires a good attitude.

Attitude toward technology: Highly experienced.

Education: Subject Matter Experts. Therefore a good level of education or experience in the domain

Linguistic skills: Very much required.

User Name: the Team Lead/Project Manager

User Role: The third category is for the Team Lead/Project Manager (depending on the size of the team and budget of the project) who is monitoring (keeping tab) on the current progress and looking ahead in the sprint calculating, how many task each member has to complete in order to burn down maximum tasks in the current sprint. He also is responsible to re-shuffle the task amongst the team, so that resources with more experience get the more complex tasks. Further, his task extends that if need arises he himself joins in executing the tasks and completing the items for the sprint.

Subject matter experience: They are generally not required to have detailed knowledge of the product but should know how to manage the team

Technological experience: Depends on the project requirement. Generally not required a highly technical manager

Other user characteristics:

Physical abilities/disabilities: Does not matter

Attitude toward job: Very enthusiastic. The job is quite tough and requires a good attitude.

Attitude toward technology: Moderately experienced.

Education: Good level of education or experience in the domain

Linguistic skills: Very much required.

User Name: Software Developer

User Role: The forth category is for the software developers who write the code and perform unit testing on the code they have written. Defects logged/Issue raised by the tester can be formed as new product backlog item.

Subject matter experience: They are required to have detailed knowledge of the product and high coding skills

Technological experience: Lots of experience in developing/scripting projects.

User Name: Database Experts

User Role: The fifth category is for the database experts who carry on with the task of storing up the data, setting up the server, deploying the product amongst the various test environments and finally providing support during the live deployment of the product.

Subject matter experience: They are required to have detailed knowledge of the product and high coding skills

Technological experience: Lots of experience in developing/scripting projects.

User Name: Client

User Role: The sixth category is of the client who basically needs the graph/report of the items that were successfully deployed/tested during the sprint, how much effort was logged in, what is the capacity of the team, how much more the team can take in the next sprint, number of showstopper defects logged in the sprint etc.

Subject matter experience: They are generally not required to have detailed knowledge of the product but should know how to manage the team

Technological experience: Depends on the project requirement. Generally not required a highly technical manager

5d Priorities Assigned to Users

Ozil is made for those users who want to develop their project following agile methodology and want to keep track of the various tasks status in while their development and generate a report either at the end of the task or have the report generated at the intermediate stages.

Key Users: These are the actual users of the organization/group/start-up who use our product for their tracking down of day to day activities in the current sprint. They can be either business analyst, testers, developers, managers, tech lead or client representatives.

Secondary Users: These can be the set of people who rate variety of similar functional products and publish their good/bad and the ugly part of the software in their

magazines/blog post. Few of them can also be unimportant users as they can be a part of smirch campaign. (Very common phenomenon nowadays)

5e User Participation

User feedback is perhaps one of the most important feature while developing the product. As, we aim to provide a better comfort to user along with a better feel and touch of the UI than our competitor, it becomes all the more useful.

We provide the users with options, to send us their feedback directly. Also, while crash, we ask the user through system interrupt to send us the report. We also sign a pact with the user which allows us to monitor the state of our application on their machines which will be quite handy in striving to make the product bug-free in the future releases.

We also provide Training Videos to the user along with tool tips which can be toggled on/off in the settings which will help the user getting acclimatized with our product.

5f Maintenance Users and Service Technicians

The first thing that we are providing to the user is a set of documents that detail where and how to reach us. The second thing that we are providing is a set of FAQs that the user can directly look into in case, he gets stuck up in the problem with our product.

There will be a team which will providing support to the user and their task is to help our customers out on the basis of ticket logged. So, basically each time the product crashes, user reports a bug, user sends a negative feedback, a new ticket is logged and immediate response is sent to the user.

5g Other Stakeholders

Testers: Testers will make sure to discuss and fix any bugs that may exist in the program. This will help keep the game updated and keep the users happy.

Marketing experts: This group is extremely important in helping spread the game throughout the world with their advertisements. It will be marketing team's job to find ways to advertise the game through online advertisements or billboards or other ways.

6 Mandated Constraints

6a Solution Constraints

Numerous times the developer will facing issues, some of the constraints that are put in place in the following section:

i. Description: The customer would want to host the software at their local machines or create a server in their local environment that can be only accessed locally.

Rationale: Since, their might be security or exclusivity related issues, many of the clients of our customer would not want their data/tasks to be stored at a third party location, hence option to host the Ozil at their end has to be provided.

Fit: In this case, customer will have to download the executables and install the software in their machines.

ii. Description: The customer would want to have access to fonts of different families along with various size, styles and shapes than is been offered in today's similar software.

Rationale: As the agile methodology is becoming increasingly popular, it is very important to target customers of various fields. Presenting them with an option of a really cool User Interface along with giving them the option to choose from an array of really cool and flexible fonts can attract the customers towards the product.

Fit: In this case, customer will be given an array of fashionable fonts to choose from. These fonts will be user customizable with the ability to change the shape, color and size of the fonts.

iii. Description: The customer would want the software to run for all the browsers

Rationale: Since, there are multiple browsers available today, the customer can choose any browser to access the software.

Fit: The software UI look and feel should be the same in case of multiple browser environment

6b Implementation Environment of the Current System

The software should be cross platform regardless of the operating system. The users will be using MS Windows, MacOS or Linux operating system on their personal computers and may not afford to get another computer in order to play the game. The product shall be approved as cross platform compliant by the QA testing group.

6c Partner or Collaborative Applications

Ozil will be multi-tier design, and the database tier will be hosted on the central server; therefore, it is important to have the client user interface with the ability to establish a connection to the database server.

Since the product requires to be cross browser compliant, it is up to the developer to see what package will be suitable. The product's UI should look and feel the same amongst the different browsers.

Also, since the product requires to be cross platform compliant, different network protocol from different operating system may be used. It is up to the developer what package is suitable to use as long as it fulfills the cross platform compatibility.

6d Off-the-Shelf Software

Ozil provides the customer with a very friendly yet cool in an an artistic fashion environment to work in. For the front end, the developer can use javascript libraries that are available in the market in the form of open source, on top of editing HTML5 and CSS, can create buttons, forms and other stuff with the help of gimp or adobe illustrator and might even use unity or other 3d animation tool if required at a very extreme end.

For the back end, data can be hosted on server locally created using python or Nodejs and for database, MongoDB or NoSQL can be used. Although, in the future, with the increase in the user data, it will be useful if the developer uses pandas to query from the server.

6e Anticipated Workplace Environment

Basically, anybody who wants to use agile system as their software development methodology should be able to use Ozil. As we are targeting both the big/established firms along with start-ups, it is not always necessary that the customer of either organizations have proper background in using the software or its ilk. Hence, we have decided to include "Training Videos" as well as small tutorials in the form of caption hints (which can toggled on/off).

Moreover, since the startups might face issues with Physical space on their machines, we provide them external servers (like cloud application) to log into the system and maintain their work. So, the task of manually installing the application on their system is now removed.

6f Schedule Constraints

Sprints are generally for 2 weeks. Although, they vary from company to company and from customer to customer.

Each release is very critical because the customer has his own critical data stored on the server. Now, the customer may have his own deployment on any day including the weekends and if it is an established firm, chances are high that the customer may be providing 24 hours service, and uses the software to update the data on the server or to manipulate/access the data in any form. Therefore, it is very necessary to inform the customer well in advance that there is going to be a weekly release and also to deliver the release within the specified date and time.

6g Budget Constraints

With Ozil, we are targeting customers of various ranges, be it in age, sex, team size, established firms and startups. It can be a 2 person team in an apartment to small startup in the valley to a giant like adobe or even Microsoft. Therefore, budgeting is very important as we don't want to overwhelm or underwhelm any customer of any size because of our cost of building and maintaining the software.

Therefore, we use the freely available libraries in the market along with free popular IDEs and different tools.

Moreover, based on the requirement of the customer, it may be possible that different people having high skillset of any single field maybe required but people with multiple skills can be deployed in the beginning and only after the customer starts paying for full version of the software or needs an exclusive customizable software as per his purpose, we can dedicate a separate team towards his requirement.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

The terms including Acronyms and Abbreviations, used in the project are as follows:

SYSADMIN: System administrator is the one who grants access to all applications, their administration features and Site administration, which includes managing users and bills

Admin: Administrator is the one who grants access to all applications and their administration features (excluding Site administration)

User: The customer or group of customers who are using the software.

Dashboard: The page where the user is able to check the tasks that are currently in his plate.

Projects: The page where the user can see and manage the task of single or multiple projects.

Issues: The Product backlog item or defect that is been created for the current sprint.

Story: The summary of the PBI or defect of the current issue.

Priority: The urgency or severity of the task or the defect (High, Medium, Low).

Edit: Modify the current task.

Assign: Assign the task to the available resource.

7b UML and Other Notation Used in This Document

Content

This section should describe the specific meaning of any symbols, punctuation, subscripts, superscripts, etc. used commonly throughout the document. If following published or common standards, then it is acceptable to reference those standards, and list any exceptions.

Motivation

If the distinction between a hollow arrow and a solid arrow is significant, for example, then everyone must know exactly what the distinctions and meanings are.

Considerations

If a particular notation is only used in one place, say on a single diagram or in a single section, then it may be more appropriate to document it in that specific location.

Example

This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in [4]. Any exceptions are noted where used.

7c Data Dictionary for Any Included Models

Create Sprint: Creating a new issue or bug. This is the first step of the workflow wherein the user can create new product backlog item or defect found for that release.

Assigned To me: These are tasks that are assigned to the currently logged in user. He can click on the Assigned To me link and check for the tasks assigned to him.

See Workflow: Clicking on this link, the user can check for the task card and their status.

Burn Down Chart: Generally for the project managers, who can check/report/verify at the retrospection meeting the number of task completed and hours taken to complete the task.

Active Sprint: This will display all the items that are available for the current sprint along with the status of the task.

Start Sprint: This will start the sprint and all the activities related to the sprint will start getting recorded.

8 Relevant Facts and Assumptions

8a Facts

Starting the sprint:

The user can begin the sprint activity by creating product backlog items through creating issues and clicking on create sprint button

Changing/Updating the task:

As and when the user changes the task status for example from “To Do” to “In Progress”, the same will be displayed in pictorial format in “view workflow” page.

Logging Defect:

The user can log defects using create issues.

Checking The Tasks Assigned to the user:

The user can check for the tasks that are currently in his plate by clicking on the Assigned to me link.

Assigning the task:

The scrum master once creates the task, can assign the same to the available user after selecting the task and assigning it to the resource. Moreover, the resource can also take up the task himself on clicking on “Assign To Me” link.

8b Assumptions

We are assuming that by the time our product will be launched in the market, the user will have enough familiarity with the handheld devices so that in the future we can release for the handheld/tablet/phablet devices too.

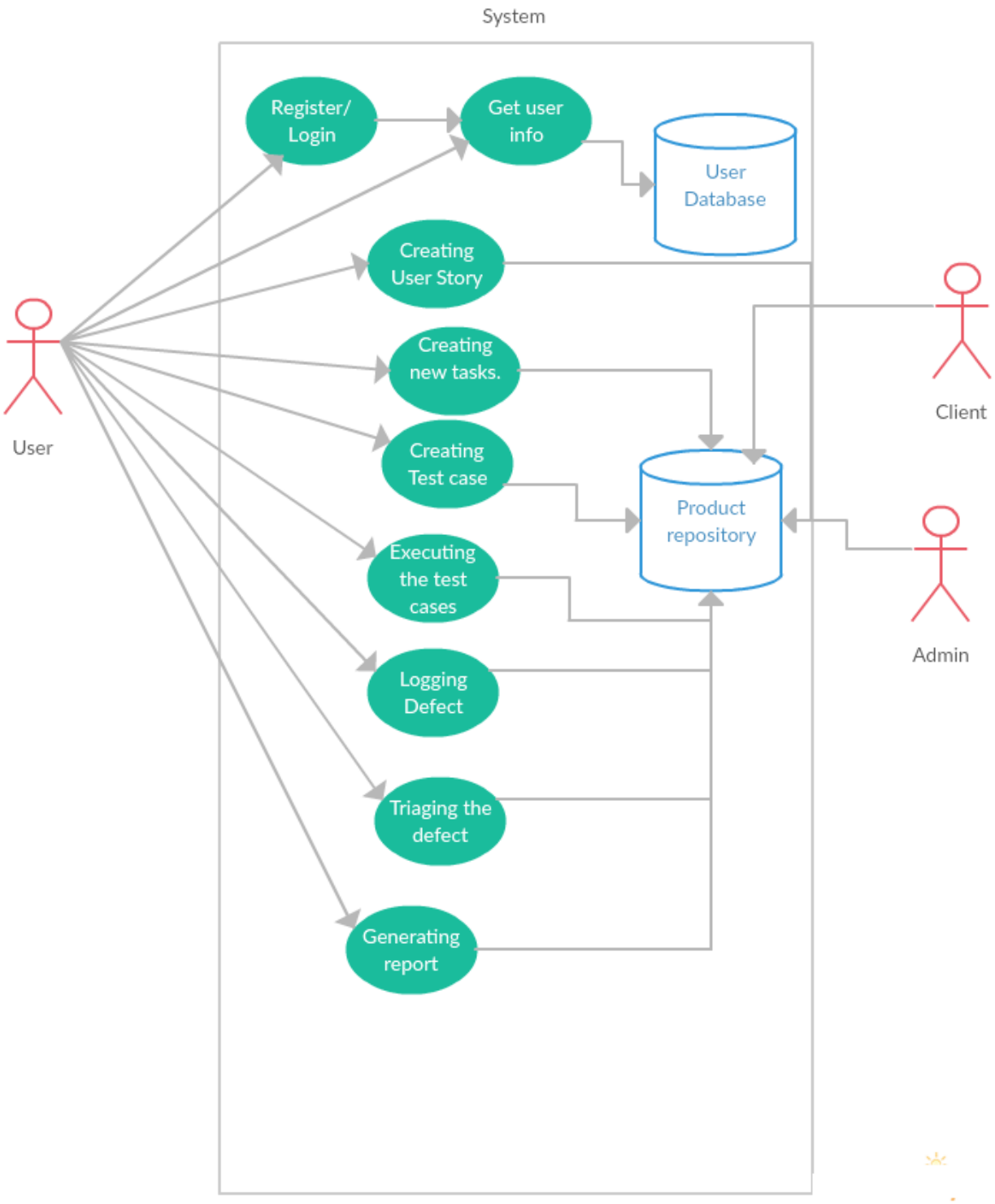
We are also assuming that smart libraries for displaying animation/smoother transition is available so that the UI can be created in an attractive fashion.

Finally, we assume that the popularity of agile methodology has increased and companies in addition to only those who provide IT solutions will be willing to part take in our product which will give us better angle to release future product and hence more space to maneuver.

II Requirements

9 Product Use Cases

9a Use Case Diagrams



Use Case Diagram

9b Product Use Case List

The following is the list of use cases, further developed in the next section.

1. Register User.
2. Login Account.
3. Creating User Story.
4. Creating Task for the use cases.
5. Creating test cases.
6. Executing the test cases.
7. Logging Defect.
8. Triage the defect.
9. Generating the report.

9c Individual Product Use Cases

Use Case Name	Register User
Participation Actors	Initiated by New User Communicates with User Database
Flow of Events	1. The system displays welcome screen with a “Sign Up” button. 2. User clicks “Sign Up” button. 3. The system displays a pop up window asking the user to enter Username, First Name, Last Name, E-mail and password. 4. User enter required information and clicks “Submit” 5. The user is brought to the welcome screen from where the user can login using the credentials.

Entry Condition	User access the application through desktop application
Exit Condition	User can successfully login using the credentials provided at the time of Sign Up.
Quality Requirements	User's information stored in the User database.

Use Case Name	Login Account
Participation Actors	Initiated by User Communicates with the User Database
Flow of Events	1. The system displays the welcome screen with a text field for Username and Password and a "Login" button 2. User enters required information and clicks "Login" button. 3. The system verifies username and password from the database and if found incorrect displays "Incorrect Username and Password" 4. If user enters correct information, the user is logged in and the system displays User's portfolio and the user can start playing the game.
Entry Condition	User access the application through desktop application
Exit Condition	User can view the portfolio and can start using the application
Quality Requirements	1. Authentication should be done within 200 Milliseconds. 2. Portfolio should be displayed first and then application may start.
Use Case Name	User Stories
Participation Actors	Initiated by User Communicates with product repository Client can interact with the product repository.
Flow of Events	1. The user logs into the system.

	2. The user clicks on create new user story link. 3. User enters details for creating the story and clicks on create button.
Entry Condition	New User story should be created.
Exit Condition	User Story created should be displayed in the User Stories section under the current sprint
Quality Requirements	The story should be saved in the product repository.

Use Case Name	Creating Tasks
Participation Actors	Initiated by User Communicates with product repository Client can interact with the product repository.
Flow of Events	1. The user logs into the system. 2. The user clicks on create new user story link. 3. User enters details for creating the story and clicks on create button. 4. User clicks on Add Tasks link and creates new task for the particular user story.
Entry Condition	New Tasks for the user story should be created.
Exit Condition	Tasks created for the user stories should be displayed in the To Do section under the current sprint

Quality Requirements	The tasks should be saved in the product repository.

Use Case Name	Creating Test Cases
Participation Actors	Initiated by User Communicates with product repository Client can interact with the product repository.
Flow of Events	1. The user logs into the system. 2. The user clicks on create new user story link. 3. User enters details for creating the story and clicks on create button. 4. User clicks on Add Tasks link and creates new task for the particular user story. 5. User logs into product repository and creates new execution sheet. 6. The user opens up the execution sheet in share mode. 7. The user starts writing in the test cases. 8. The user saves the sheet.
Entry Condition	The sheet should be saved in the product repository.
Exit Condition	On opening up the sheet, the user should see all the changes made to the execution sheet across the team.
Quality Requirements	The sheet should be saved in the product repository.

Use Case Name	Executing the test cases
Participation Actors	Initiated by User Communicates with product repository Client can interact with the product repository.
Flow of Events	1. The user logs into the system. 2. The user clicks on create new user story link. 3. User enters details for creating the story and clicks on create button. 4. User clicks on Add Tasks link and creates new task for the particular user story. 5. User logs into product repository and creates new execution sheet. 6. The user opens up the execution sheet in share mode. 7. The user starts writing in the test cases. 8. The user saves the sheet. 9. The user starts executing the test cases. 10. The user changes the status of the test cases he has executed. 11. The user saves the sheet.

Entry Condition	The sheet should be saved in the product repository.
Exit Condition	On opening up the sheet, the user should see all the changes made to the execution sheet across the team.
Quality Requirements	The sheet should be saved in the product repository.

Use Case Name	Logging the defects	
Participation Actors	Initiated by User Communicates with product repository Client can interact with the product repository.	
Flow of Events	1. The user logs into the system. 2. The user clicks on create new user story link. 3. User enters details for creating the story and clicks on create button. 4. User clicks on Add Tasks link and creates new task for the particular user story. 5. User logs into product repository and creates new execution sheet. 6. The user opens up the execution sheet in share mode. 7. The user starts writing in the test cases. 8. The user saves the sheet. 9. The user starts executing the test cases.	

	10. The user changes the status of the test cases he has executed. 11. The user saves the sheet. 12. The user fails some of the test cases that do not perform as expected. 13. The user clicks on Add defect against that failed test case. 14. In “Add a defect” page, the user gives in the steps to repro the defect. 15. The user writes in the steps to repro the defect. 16. The user attaches images and videos as a proof to repro the defect. 17. The user gives severity and priority to the defect. 18. The user assigns the defect to the concerned developer. 19. The user submits the defect.	
Entry Condition	The defect should be submitted and unique Defect ID should be generated for the defect.	
Exit Condition	The defect should be attached to the failed test case and all its attachments and defect contents should be valid.	
Quality Requirements	The defect should be saved in the product repository and all its attachments and defect contents should be valid	

10 Functional Requirements

Requirement #: 1

Description: The system shall provide meanings for the user to register and create new account with username and password.

Rationale: To be able to know the identity of the user, monitor and store their record in data repository.

Fit Criterion: A new user is provided with an option at the login menu to create a new account, where user enter their name, email, password recovery question, school and class/course to create a new user account. Once user successfully creates an account, they can login to the application with their username and password.

Requirement #: 2

Description: The system shall provide an option to recover password for existing users in case if they forget their password.

Rationale: Recover password and login to the application.

Fit Criterion: An existed user can recover their password by answering the security question they selected and answered at the time of their registration. Once they recover their password they can successfully log back into the application.

Requirement #: 3

Description: The user should be able to create user stories in the application.

Rationale: To be able to create and store user stories in the application.

Fit Criterion: A user is allowed a section under current sprint wherein he can create user stories and view all the created user stories. They are fundamental building blocks of the development lifecycle as they contain the set of tasks which are to be performed by the user in the current sprint.

Requirement #: 4

Description: The user should be able to create tasks in the application.

Rationale: To be able to create and store tasks in the application.

Fit Criterion: A user is allowed a section under current sprint, under user stories wherein he can create tasks and view all the created tasks. He can assign and reassign the tasks to the different members of the team from this section.

Requirement #: 5

Description: The user should be able to create test cases in the application.

Rationale: To be able to create and store tests cases in the application.

Fit Criterion: A user is allowed to log in to the product repository server and it is a "single sign in" functionality wherein he will be allowed to create execution sheet, give name to the sheet and open it in "Private" or "Shared" mode. The user can enter his test case execution details in the sheet and save the sheet back into the repository so that other users can access the sheet and see the modifications in it.

Requirement #: 6

Description: The user should be able to sync test cases in the application.

Rationale: To be able to sync the tests cases in the application.

Fit Criterion: The other way for syncing in with the test sheet is that if the user has checked out a version of sheet say 42 on his local machine and if another user has checked in different version of the same sheet, then there is sync button, clicking on which, the user will be able to sync his version and see all the changes made by the other user.

Requirement #: 7

Description: The user should be able to execute the test cases in the application.

Rationale: To be able to execute the tests cases in the application.

Fit Criterion: The user should be able to open the execution sheet and change the status of the test cases as "Pass", "Fail", "Blocked", "Not Applicable", "Can be checked Later" along with the date of execution, and comments if applicable.

Requirement #: 8

Description: The user should be able to log defects against failed test cases in the application.

Rationale: To be able to log defects against failed tests cases in the application.

Fit Criterion: The user should be able to log defects against failed test cases. So that there can be a record maintained for the test case execution effort as well as legacy of known and fixed defects can be maintained. The developer is notified directly with the priority/severity of the defect as well as steps to reproduce it along with attached documents which act as proof to serve the claim.

Requirement #: 9

Description: The user should be able to open up burn up/down graphs and charts.

Rationale: To be able to open up burn up/down graphs and charts in the application.

Fit Criterion: The user should be open up burn up/down graphs and charts. This functionality is limited to high clearance users only like the managers. They can track down the efforts put in by each individual in the current sprint and gives overall idea where is scope of improvement which can be really handy during sprint planning meeting and retrospective meeting.

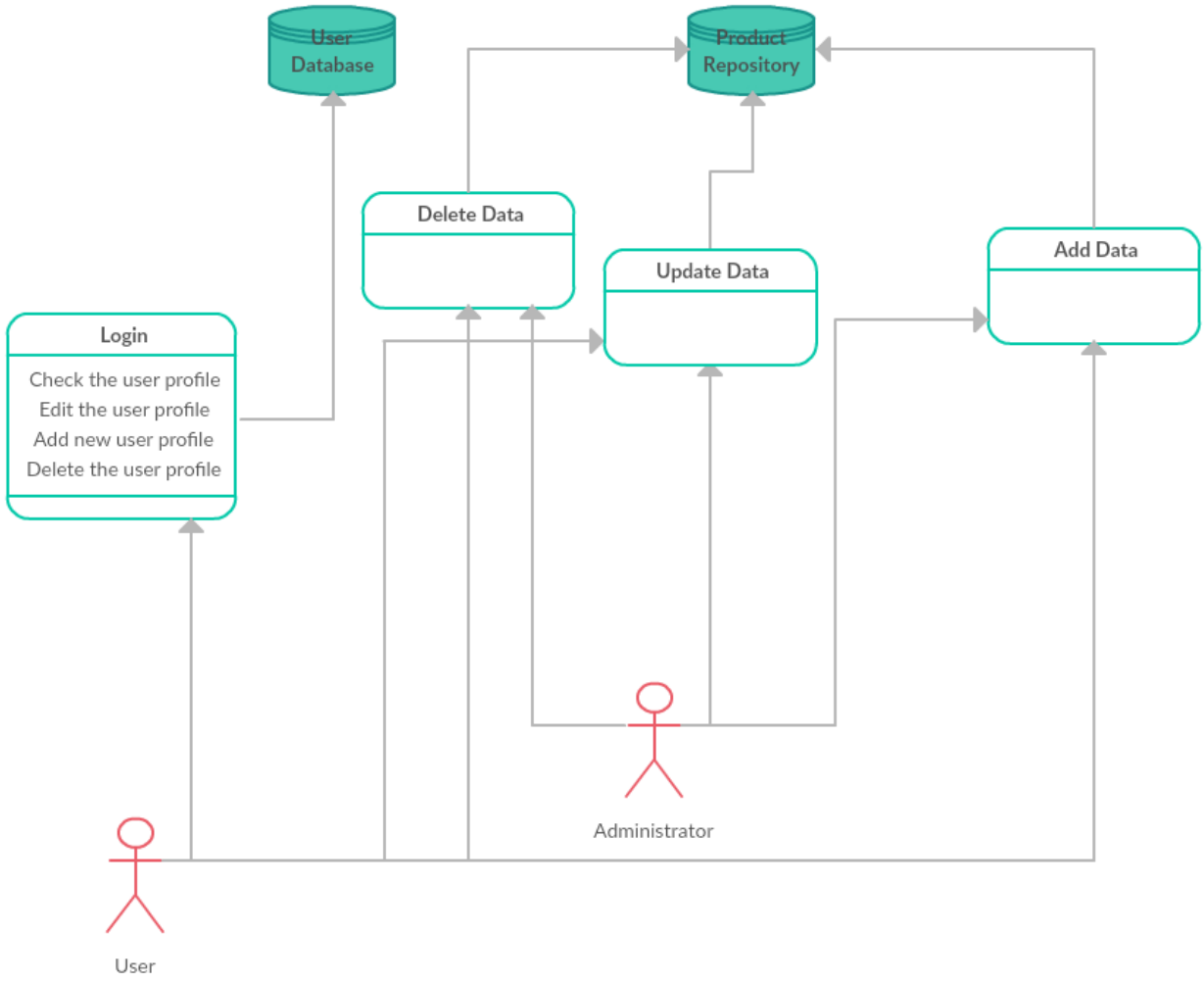
Requirement #: 10

Description: The user should be able to move the task cards along different execution status.

Rationale: To be able to move the task cards along different execution status.

Fit Criterion: The user should be able to move the task cards along different execution status. This signifies what is current execution status of the particular task and how much more task is yet left to be completed. Also, the user can by an overview gauge if the task can be moved onto any idle resource in the team so that efficiency can be increased within the team.

11 Data Requirements



Database design

12 Performance Requirements

12a Speed and Latency Requirement

- The screen must be loaded within five seconds after running the application.
- New screens and pop-up windows must be loaded within 1.5 seconds, and no response time shall take longer than five seconds.
- After the system receives login credentials from the user, returning users must be able to login within one second.
- The user's profile and preferences must be loaded within 500 milliseconds.

- System must load the user's burn down chart within 1 second.
- System must open the defect report or the test case sheet within 500 milliseconds.
- System must generate a user profile when required within 200 milliseconds.

12b Precision or Accuracy Requirements

- The users should be able to collaborate their work (i.e. the work of different users should be saved and tall the other users should be able to see the saved work).
- The same works for both the test cases and the defect entries that any of the user makes
- The number of defects logged in the system and the number of test cases executed by the user along with the number of test cases as successful passed and failed and defect id for the failed test cases should be mapped appropriately in the system.
- The burndown chart should be appropriately updated for the different users according to the hours entered by each the user and available across the team.
- The document should be properly loaded from the repository and multiple copies should be made periodically so that the user can roll back to previous version.

12c Capacity Requirements

- The server must be able to handle up to 500,000 active users logged in simultaneously.
- The server must handle at least 1,000,000 requests running simultaneously
- The server must be able to store up to 500,000 user profiles.

13 Dependability Requirements

13a Reliability Requirements

- No user data such as login credentials or virtual account shall be lost in the event of a system failure.
- The burn-down chart, linking between the menus, loading up the test case execution documents, defect reports must not experience any glitches.
- The server must not experience any failures more than once a month.
- Maintenance must be done periodically.

- The system must notify users of the time and duration of the maintenance well in advance and also on the time of maintenance should display appropriate message notifying the user that the “site is down for maintenance” and will be up in x amount of time.

13b Availability Requirements

- The application must be available 24 hours a day, 365 days a year except when doing maintenance.
- The system must allow users to access their portfolio at all times even during maintenance periods.
- Updates for the application of the must be available once every 2 months.
- New versions of the application must be available once a year.
- The product must be functional and available 99 percent of the time.

13c Robustness or Fault-Tolerance Requirements

- The system must notify all users if the database connection goes down which may restrict users from being able to log back in.
- In the case of lost internet connection, the system must allow users to save their current progress and return back to the work wherever they left off as soon and they get back internet connection.

13d Safety-Critical Requirements

- Working continuously for hours on the application must not cause any physical damage to the users.
- The application must use a well-balanced color scheme so as not to disturb players of cause them any eye injuries.
- The application must use smooth transitions whenever necessary so as not to surprise the user with a sudden and drastic change in color.
- The system must be designed with efficiency so as not to cause the user’s computer processor to overheat.

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

- All reviews and suggestions for improvements that are given by the users in the suggestion box shall be reviewed and considered by the maintenance team

- If the application site is down for any reason, the maintenance team shall have it up and running within 1 hour of the exact time they are notified of the problem.

- There shall be a member of the maintenance team on duty at all times. Support is one of the most important areas of our concern.

- The product shall be maintained by the maintenance team to ensure that all the bugs discovered after the releases are fixed promptly.

- Any patches for bugs shall be released within 10 days of the day that the bug is found.

14b Supportability Requirements

- The system shall allow quick and easy fetching of from the central database

- The product shall provide assistance through email/telecommunication for users at all times. There will be separate maintenance teams for both emails and telecommunication.

- The users shall receive the reply to their email within 24 hours.

- In the future, we may develop AI for the answering machines so that user does not need to contact the representative each time and menial issue can be solved by the AI itself, which will further reduce the cost of operation.

- If the user forgets their identification, the product shall email the user their identification information through email, provided that they correctly answered a security question.

- We will also provide the user the opportunity to reset his/her password over the phone wherein they will have to just provide with their login Id, email and through registered phone we can track the user down.

- In future, to remove the whole problems with username/password thing, we can provide password generating devices to users by which new passwords will be automatically generated for them periodically. The user will have to login with the newly generated password.

14c Adaptability Requirements

- The product shall be compatible with the current as well as 2 generation older versions of Windows, Mac OS and Linux OS.

- All features of the previous versions of the product shall be compatible with the latest version.

14d Scalability or Extensibility Requirements

- The product shall be able to support nearly 50,000 users during the first quarter of the first release.

- After one year, the application shall be able to support 100,000 users.

14e Longevity Requirements

- The application server should be available all time for the user to access data from it.

- The features of the application shall continue to improve over time, removing any bugs when they are discovered and introducing newer, better features to enhance experience working with it.

15 Security Requirements

15a Access Requirements

- The maintenance team (developer and tester) and the developer team has access to the source code.

- User's personal information such as username, password and email addresses shall not be accesses by anyone other than the user.

- The user database, team database and current product database that the team is working on (back-end) shall only be accessed by the admin level users and the leader of maintenance team. The maintenance team if handling data of different clients, should be exposed to the clients. The purpose is that client would not want their data to be in the hands of unknown entity.

- User's portfolio can be viewed by the base level users and admin level users.

- At any time, different users of different teams should not be able to view the product/team details of different teams.

- At any time, different users of different teams should not be able to access the product/team items of different teams.
- For storing and retrieving documents in GIT or any other central repository, the user has to be logged in all the time with the login id created to access the application.

- Only manager and user with high level clearance should be able to see the burn down/up charts.

15b Integrity Requirements

- The data in the user database shall not be manipulated or made available to public at any time.

- The data in the product database regarding to the product that the team is working on should not be made available to public at any time.

- A concerted effort shall be made to protect user data and product data and no data shall be lost at any point.

- The maintenance team shall also conduct weekly checks to ensure that none of the data has been manipulated or modified.

- The product shall protect itself from outside users or hackers, attempting to manipulate the data.

- All of the user data shall be backed up in a separate database to ensure safety. The backups shall take place every week to ensure that the backup database is up to date.

15c Privacy Requirements

- At the beginning of signing up for the application, the user would be given a privacy policy document to read. The user can read the whole document and accept the policies and click on the continue button or reject it on the account of which he won't be able to continue with the registration procedure. The same document will be made available in the "Policies" section.

- The product shall notify the users through email and through notification on the sign in page if any changes are made to the privacy policy.

- The user can only see which all tasks has been assigned to which users and their current status on the task. No other information about other users shall be made available to other users.

15d Audit Requirements

- The user will not be allowed to download or upload any material through his/her work machine. Any attempt to do so should be tracked by the security official with the IP address of the user.

- The user will not be allowed to insert pendrives/diskdrives/flobs in the machine without proper authorization from the security team.

- The user's antivirus has to be updated all the time in case to protect the machine from harmful attacks from outside.

- The information about all the users are retained in order to track the users in case if any mishap happens.

- The information is retained only for security purposes so that the user may not deny having used the product.

- The information includes the name, e-mail id of the user and also the time intervals when the user has actively used the product.

- Any update/edit made to the system is stored in a separate database along with the timestamp.

15e Immunity Requirements

- The application shall scan all entered data against the published definitions of known computer viruses, worms, and Trojan horses.

- The application shall delete the infected file if it cannot disinfect the infected data or software.

- The application shall daily update its list of published definitions of known harmful programs

- The application shall notify a member of the maintenance team if it detects a harmful program during a scan.

16 Usability and Humanity Requirements

16a Ease of Use Requirements

- The application shall not take more than five seconds to start although the loading of the application page depends upon the speed of internet connection.

- On loading the files from the repository, the files should be loaded with the message as of "file opening" or "file loading". The same applies for loading the different burn up, burn down charts. The user will be specified whenever and wherever the loading of the application is taking more than normal time.

- The application shall provide an intuitive graphical user interface that the intermediate user can navigate without having to read intense documentation.

- The new user should be provided with hints how to navigate the system. The experienced user shall have an ability to refer to more information menu regarding how to navigate and use the application effectively and efficiently.

- There would tool tips present on top of the icon, help file in the help menu and also video tutorial on how to use the application

- The user shall have an ability to recover their password by answering their security question and getting their password in email listed on their account.

- The application shall provide correct and up to date information when the user is connected to the network from the main database server. In an event when system is not connected to the main database server, they should be provided with feedback of the last time when the information currently displayed was fetched from the main database server.

16b Personalization and Internationalization Requirements

- The application will be initially available in English language and later with the increase in popularity will be available in different languages too.

- The user should have ability to choose a region or country of their preference in the later version of application.

- Initially, the help and supporting documents will be in English language but as and when new languages will be available, both the documents and videos should be available in respective languages.
- The tool tip will be displayed in the choice of the language selected.
- The charts and graphs will also be displayed in different language.

16c Learning Requirements

- Anybody who has the basic knowledge of AGILE methodology will find the application easy to use as it is made very intuitively.
- Engineer should know basics of assigning tasks, moving the tasks across various development milestone and what all steps are required to log defects and attach proof of procedure.
- The user should know how link the execution sheet to the repository and query the same from the repository.
- The manager level user should know how to bring up the correct charts and what parameters to put in for getting the correct chart.
- The application shall provide a tutorial for novice or new users to understand how the application works in effective and efficient way.

16d Understandability and Politeness Requirements

- The application shall not lock the user computer.

- The application should always prompt the user with “please” prefix for their inputs.

- The application at any point of time shall not include any violent graphical images.

- The application shall not use any word that can be offensive to any member of the community.

- The application shall not use any technical developers' jargon in the information or prompts. The language used for either of these should be layman or everyday spoken common word, so that the users are not confused.
- The same applies to the tool tips and the help document too.

16e Accessibility Requirements

- The application shall be accessible by all individuals as long as they can access a computer, a browser and internet connection.

- The application shall conform to the Americans with Disabilities Act to be qualified as learning tool used by schools or universities.

- The application shall provide a conversion tool for different colors to set of user preferred colors, in case of individuals with color blindness.

16f User Documentation Requirements

- The application shall provide an electronic technical specifications to accompany the product with ability to print.

- The application shall contain user manuals for reference in electronic format with ability to print.

- The application shall provide a service manuals for administrators in printed and electronic format with ability to print.

- The application shall provide emergency procedure manuals in case of data outage for the administrators.

- The installation manuals should accompany in electronic format with application installer for ease of use and step by step installation guide for the user.

16g Training Requirements

- The administrators of the client's product database will require a one to two day training by technical support staff of the application to make sure administrator understand how to perform various admin related tasks for example, backing up

database, recovering lost database, updating the database and providing support during release of the product.

- The application will provide new user tutorial with step by step video to guide the new users.

17 Look and Feel Requirements

17a Appearance Requirements

- The application has to be styled in the latest technology as has been stated in the above section.

- The application will allow the user to choose from a set of multiple themes and apply to the application and moreover they can choose from a range of fonts as well. Thus the application is flexible enough to meet different users.

- The application shall be attractive to users of all ages.

- The application shall not display any images deemed to be inappropriate.

- The application shall not display any content deemed to be inappropriate. This includes, but is not limited to, profanity and any other inappropriate words or phrases.

3a Style Requirements

- The product shall make a user comfortable and at home and even though he might be well versed with the concepts of SDLC and AGILE methodology, everything should be at his finger-tips which means that the menus and icons should be placed or categorized in such manner that the frequently and same functional items are grouped in one menu.
- The applications such as bringing up the graphs and reports should be well animated, the dragging of the cards across the various milestones should properly done so that it looks attractive on the eyes.
- The menu and icon and other text should be appropriately large so that the user can easily read whatever is written on the screen and easily navigate across the different pages in the application.
- The application's menus should look simple and avoid clustering a lot of materials together.

18 Operational and Environmental Requirements

18a Expected Physical Environment

- The application shall be used by a user on their computer, in office environment be that in a building or in backyard of their house. So the noise decibel level will depend on wherever the office is set up.

- The product can be accessed on a desktop or laptop that fit in a backpack or purse that can fit their laptop.

- The application shall be usable in dim light or brightly lit room.

- The application shall be accessible at any time of the day with exception of pre-scheduled down times.

18b Requirements for Interfacing with Adjacent Systems

- The application shall work on the last two releases of supported operating system at the time of release.
- The application shall have the capability to reestablish a lost connection when the system connects back to the network.
- The application installed on the user's computer must be able to ping the database server.
- The application shall be able to access the database server either through internet or intranet.
- The local database installed on the user computer should be configured through the installed application on the user's computer.
- The administrator should be able to configure the database server through the application interface rather than having to interact directly with the server computer.

18c Productization Requirements

- The application's online version shall be accessible only online and the offline version is available to setup on the user machine only after they have purchased the license and the license is not redistributable.

- The application's offline version shall be able to be installed by double clicking on the setup file by following step by step prompts without separately printed instructions.

- The application's offline version shall contain all pre built binaries of all the prerequisite required for the application to function in the setup file without relying on internet to fetch those prerequisites.

3b Release Requirements

- There shall be a maintenance release periodically that will be offer all new features and accumulated bug fixing patches from the previous releases.

- All previous features shall not be affected by the new release.

- The maintenance team require to provide a hotfix patch in timely manner for any identified bug.
- A new release require thorough testing by QA before it can be released to the user.
- Every new release will contain release notes that will explain the new features and bug fixed in the release.

19 Cultural and Political Requirements

19a Cultural Requirements

- The application shall not purposefully intend to offend individuals of any particular culture.

- The application shall be released in multiple languages, so as to support multiple regions and cultures.

- The application shall not favor any particular group or ethnicity.

19b Political Requirements

- The application's policy shall adhere by all laws laid out by the constitution of the United States

- The application shall not mock anyone's race or ethnicity or religion or place of birth or include harsh language in the application or in the help menu.

20 Legal Requirements

20a Compliance Requirements

- The application's offline version shall not distributable user information to third-party source without the consent of the user, thereby complying with the Data Protection Act.

- The application shall avoid all copyright infringements.

- The application's online version is initially available for free and later the license has to be purchased for getting the full features of the application.

- The product shall comply with industry's rules and regulations.

20b Standards Requirements

- The product shall meet the standards set by the Federal Government pertaining to the Information technology industry.
- The product shall be developed following the agile development process.

III Design

21 System Design

- During the system design, the system is to be transferred from the analysis model into a system design model. We have defined the design goals of the product and decomposed the system into smaller subsystems.

21a Design goals

- The design goals represent the desired qualities of the Ozil Product, and it represents a consistent set of criteria that must be considered when making design decisions. Our main purpose is to deploy robust, maintainable, well-designed, and reusable software with object-oriented analysis and design. We are determined to define and visualize each and every perspective of the system explicitly in order to completely materialize out object-oriented approach. Next, we also pay attention to how to diminish the influence and impact of alterations or security exploits.
- The design goals identified in details are as follows:
    - Adaptability: Java is a popular programming language in terms of providing cross-platform portability. Therefore, it would be an optimal choice for the development of our program. Java enables our system to work in all Java Runtime Environment (JRE) installed platforms; therefore, the user will not have to worry about the operating system requirements. In order to fulfil this adaptability feature, we suggest programming the game in Java.
    - Efficiency: The system is going to be responsive and be able to run with high performance.

- Reliability: The System will be released bug-free consistent in the boundary conditions. Any bugs that might be discovered after the product is released shall be maintained and fixed by the maintenance team. The system must not crash with incorrect inputs.
- Usability: Easiness in the usage is an important design goal in product as well as any successful software system in terms of user's comfort. It makes the product more friendly and attractive. Therefore, the system will be designed such that user can easily interact with the system without any prior knowledge. However, user-friendliness of the system does not imply making the gameplay easier, which might make the player bored sooner than expected.
- Extensibility: Object oriented architecture of the product enables system customizations without causing any bugs during modifications.

22 Current Software Architecture

There is no current software architecture; however, bellow is the proposed software architecture.

23 Proposed Software Architecture

23a Overview

The architecture model will compose of three tier design for Ozil project. It entails user interface layer, the application business logic layer and the storage layer for data bases. The User Interface will provide a means for the user to interact with product. It will display appropriate information for the user to make progress. The application business logic layer contains the logic to check against if the user information entered is correct, and it will also validate the user’s retrieval information. The data access layer will deal as interface for the application to interact with data base management systems. The storage layer also contains up to date DBMS server address path along with require credentials to connect to the remote server to fetch and update with data bases. The User interface layer shall not interact with database server, hence it must go through business layer and business layer will connect through data base access layer in order to retrieve or update information with DBMS server. The three layer model is shown below:

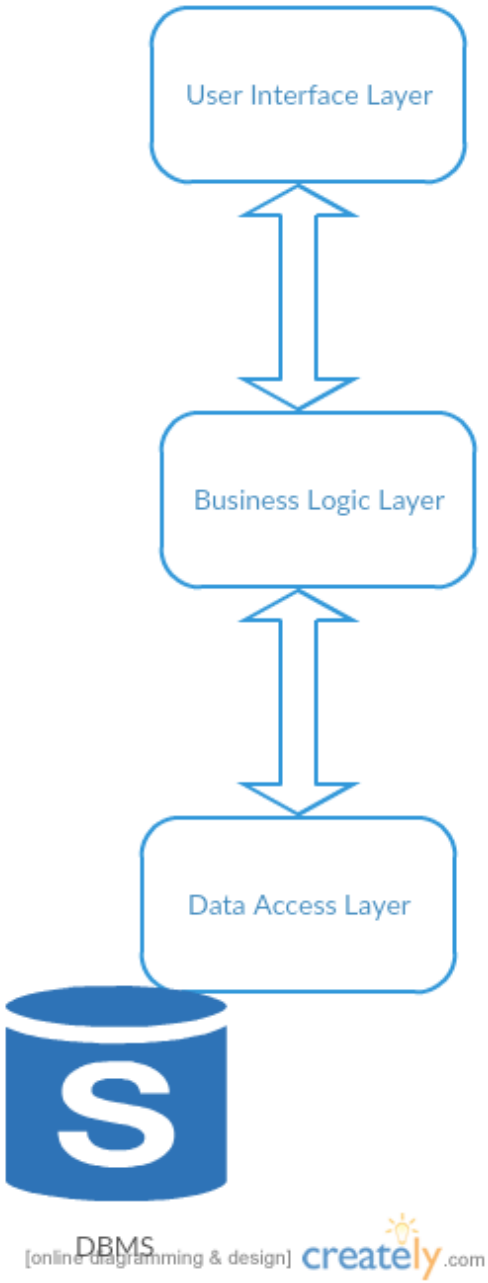
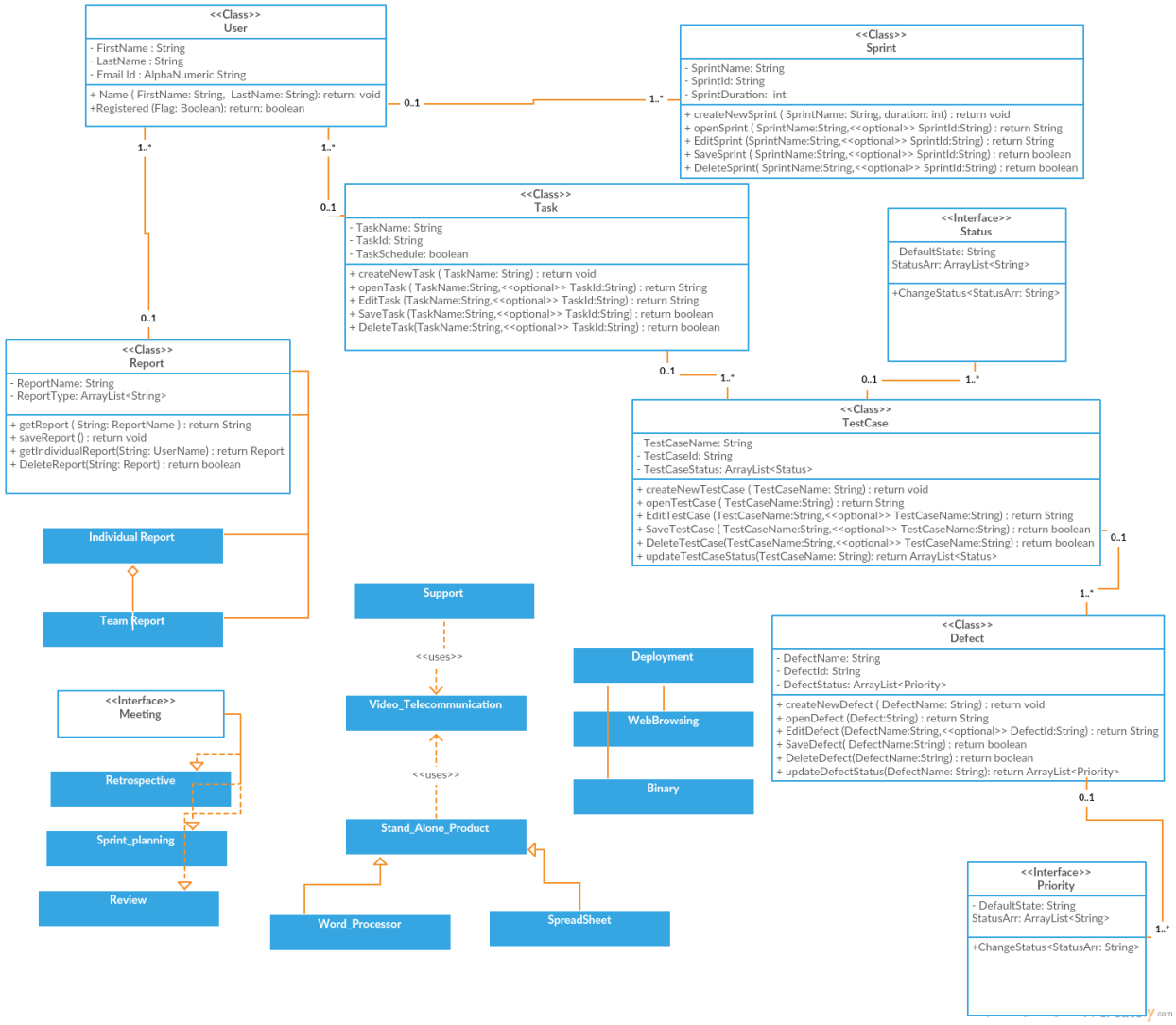


Figure Three tier architecture diagram

23b Class Diagrams

By large Ozil contains three design patterns as mentioned in details below:
1: Factory Method Design Pattern
Class User provides a UserType() method for various inherited user object creation. It checks whether the user is new or return type.
Also, the report checks whether the report is Individual or team.
2: Strategy Design Pattern
Meeting is made interface, so at run time, the user can select whichever type of meeting he is interested in and automatically all the parameters of meeting are inherited by the child object.
Deployment is of either Web version or the binary.
StandAlone Class has two subtypes Word Processor and SpreadSheet. So both these classes have all the methods and parameters at run time.



23c Dynamic Model

The sequence diagrams for each use-case are diagramed below

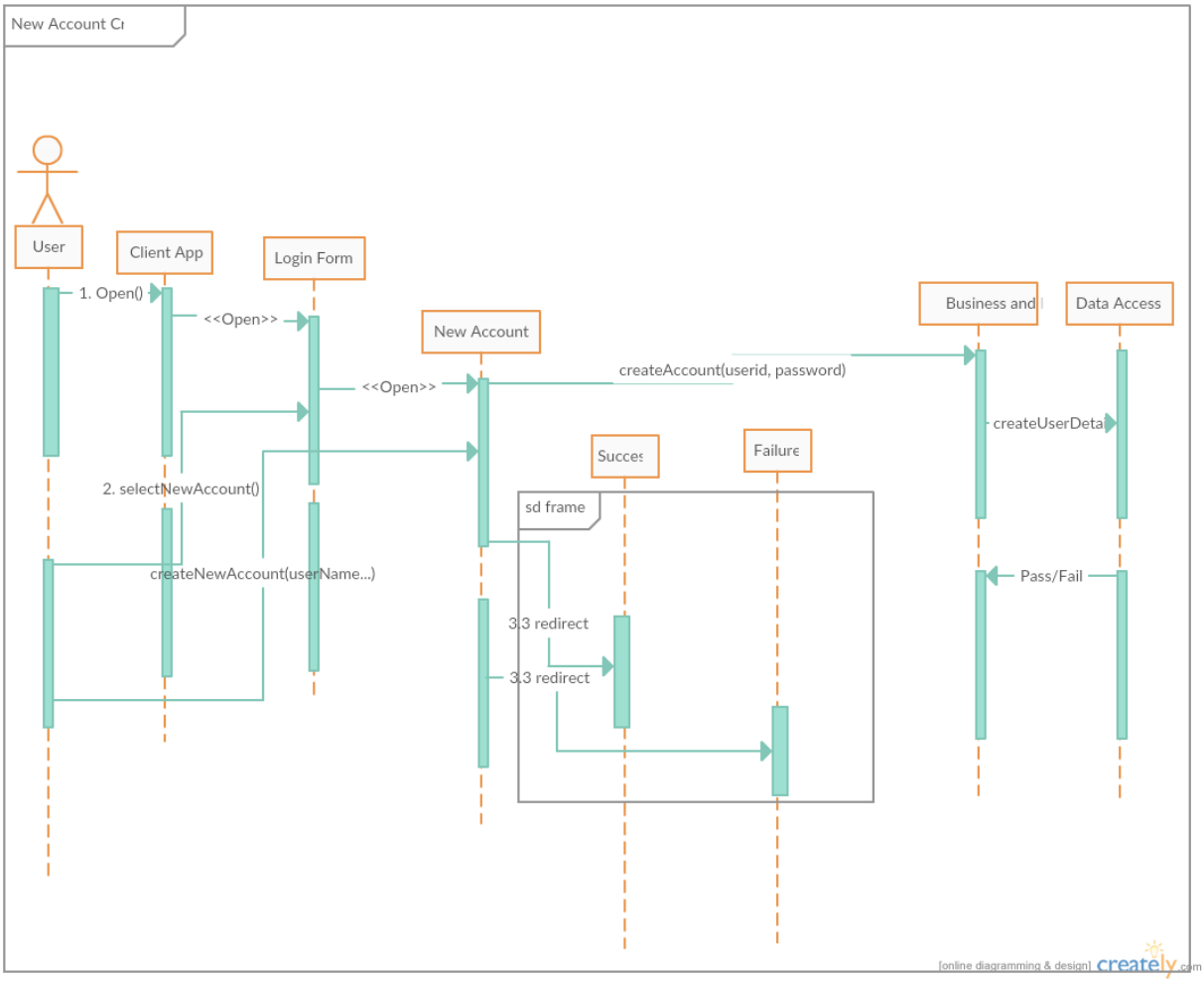


Figure New Account Creation

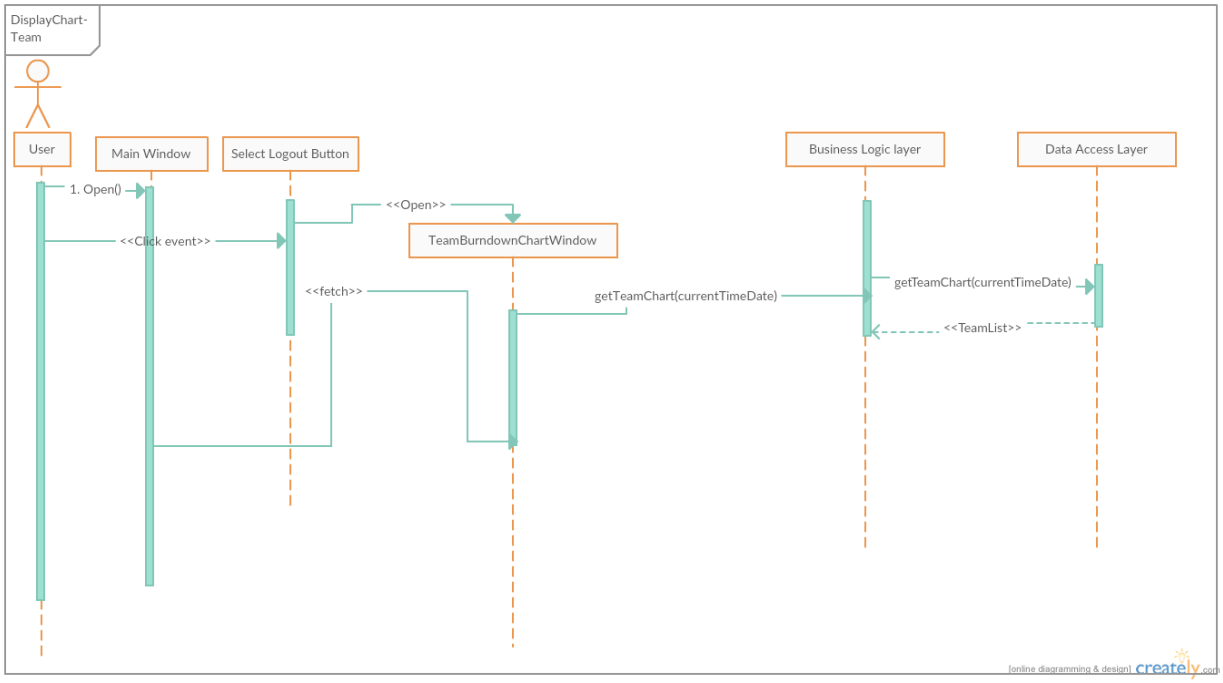


Figure Display Team Chart

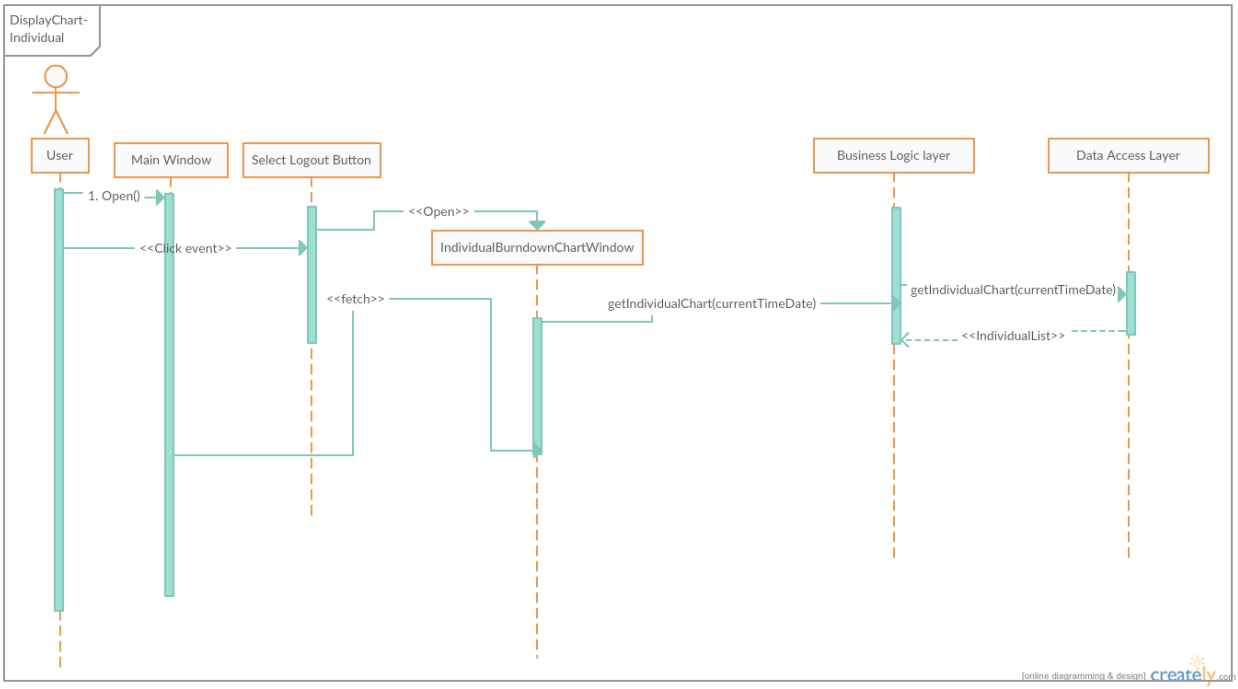


Figure Display User Chart

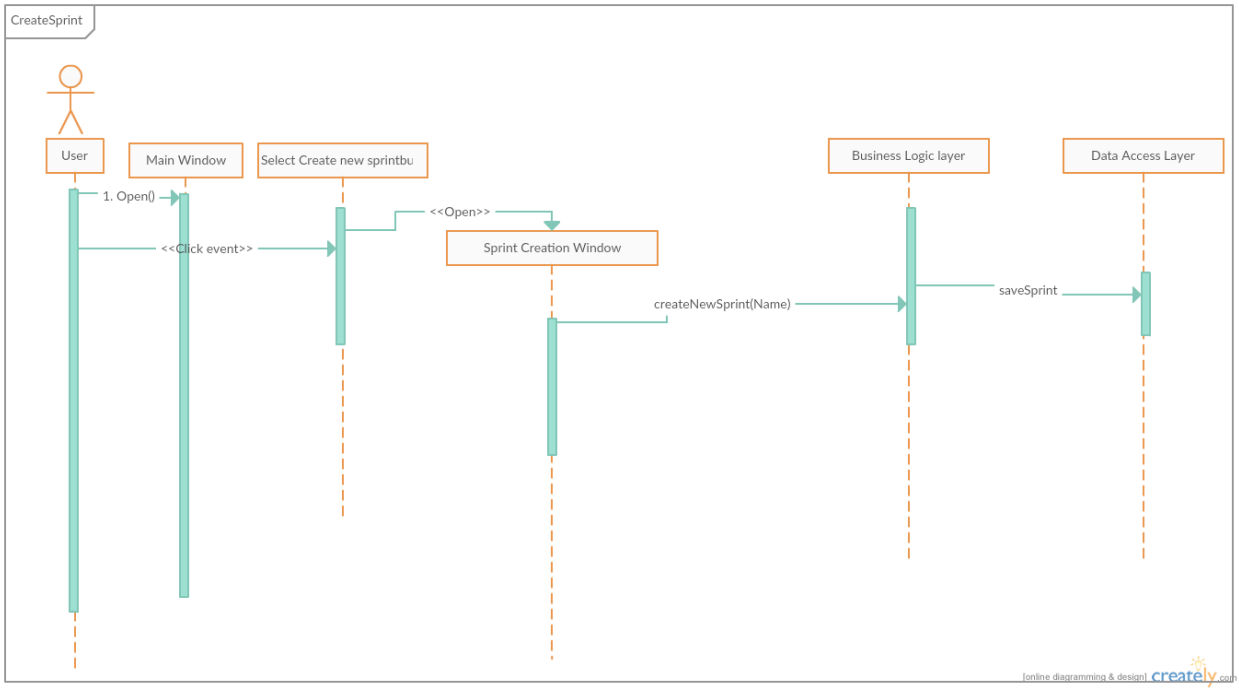


Figure Create New Sprint

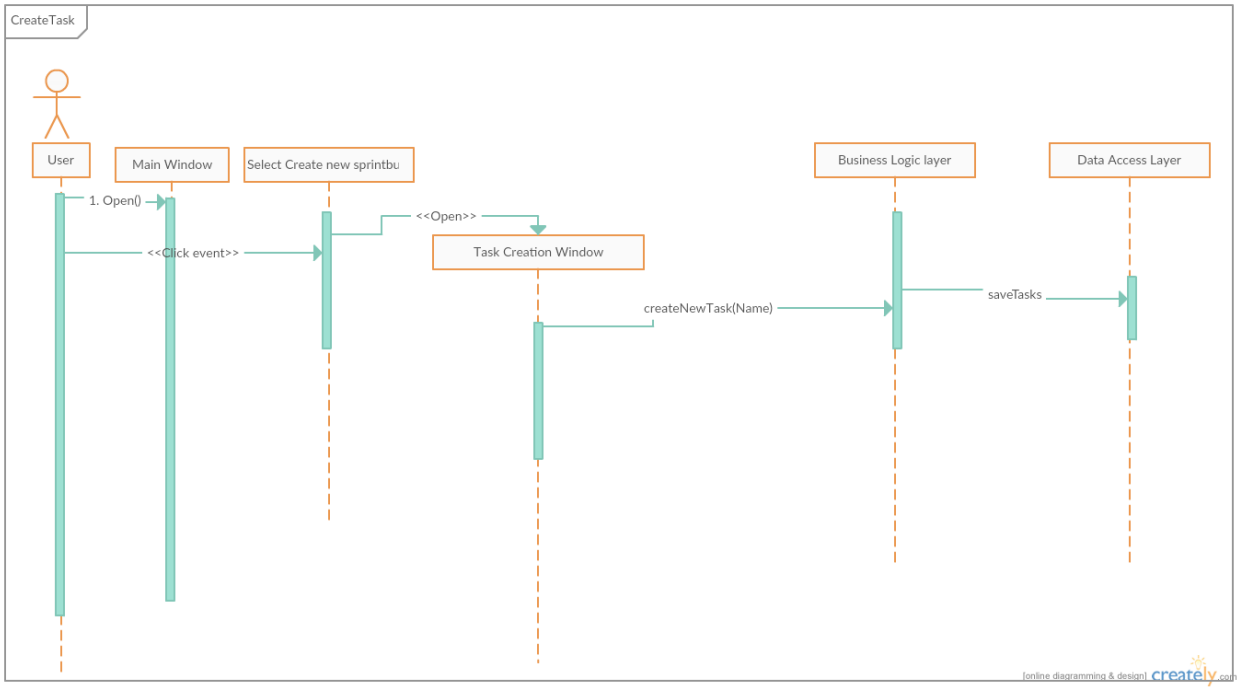


Figure Create New Task

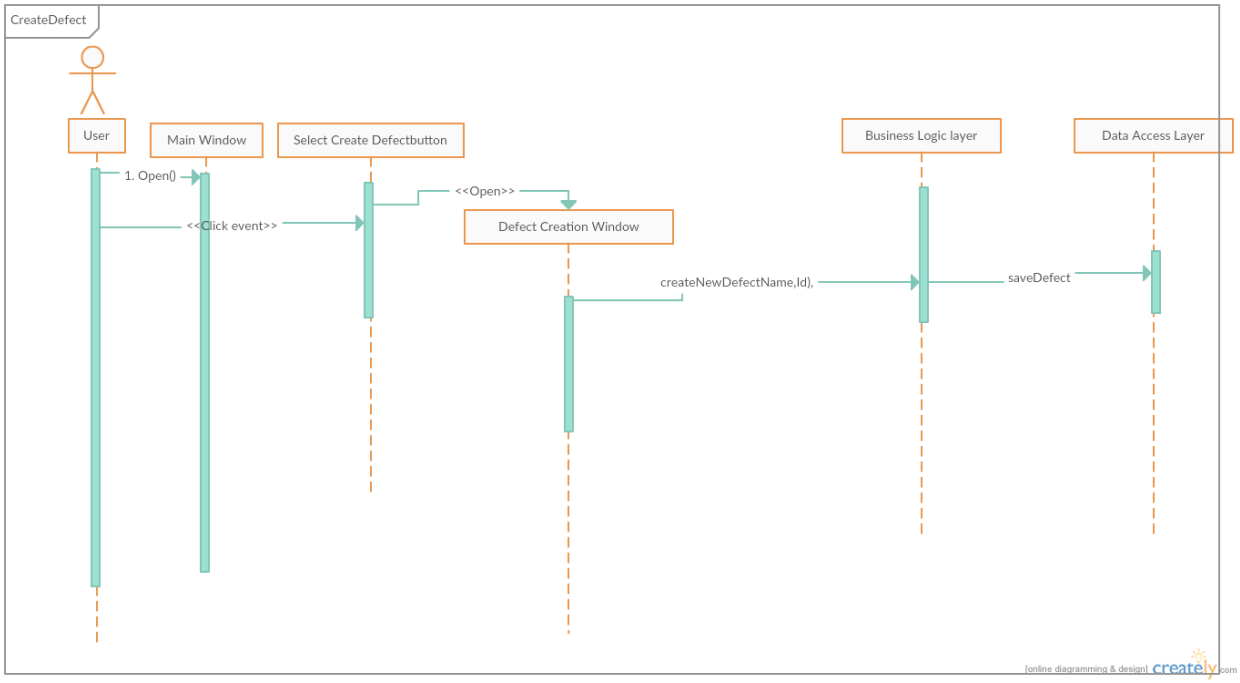


Figure Create Defect

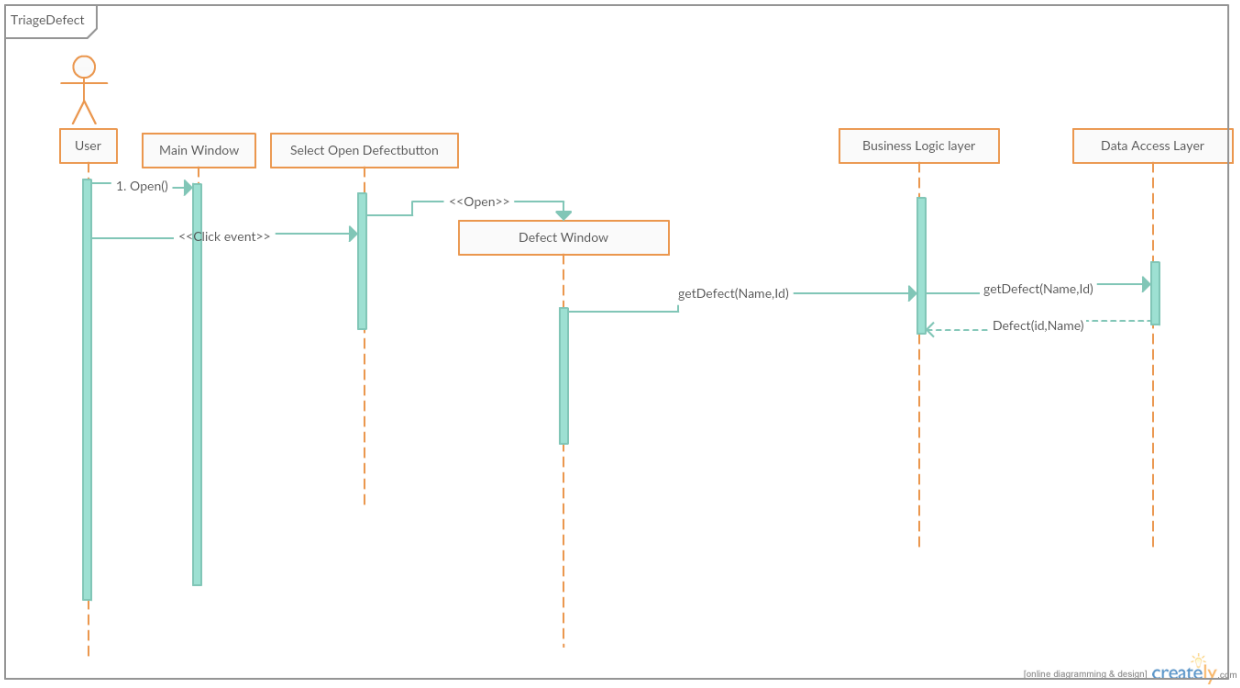


Figure Triage Defect

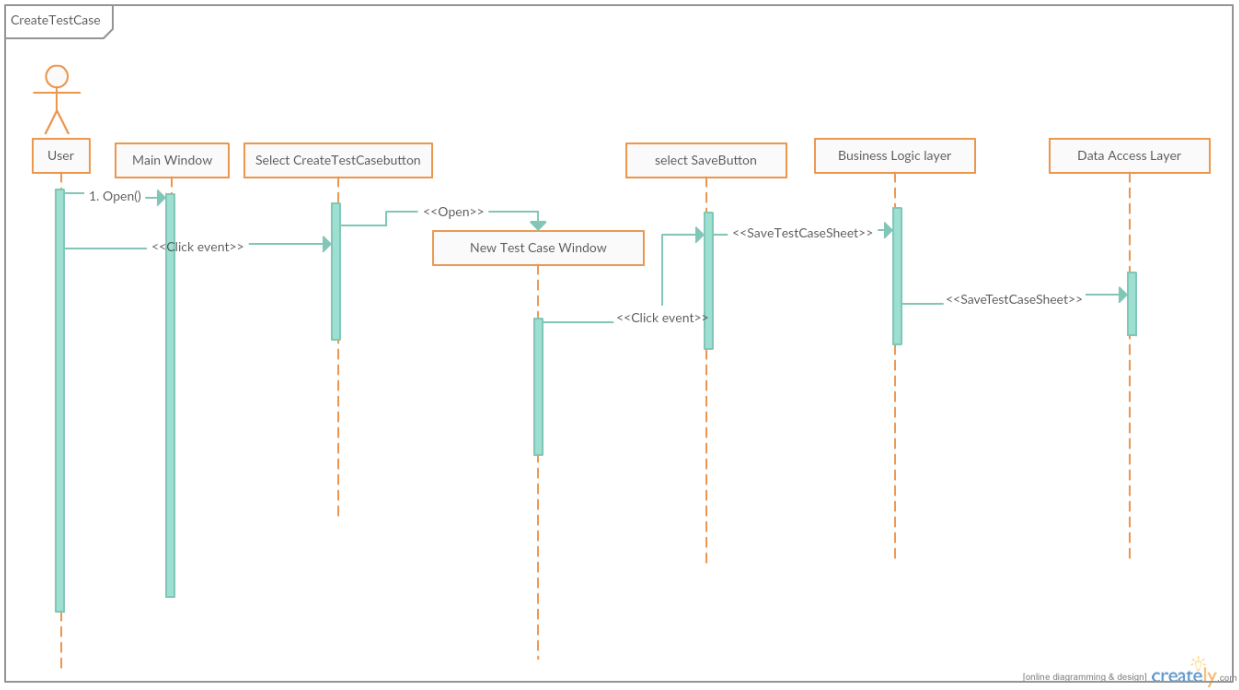


Figure Create Test Case

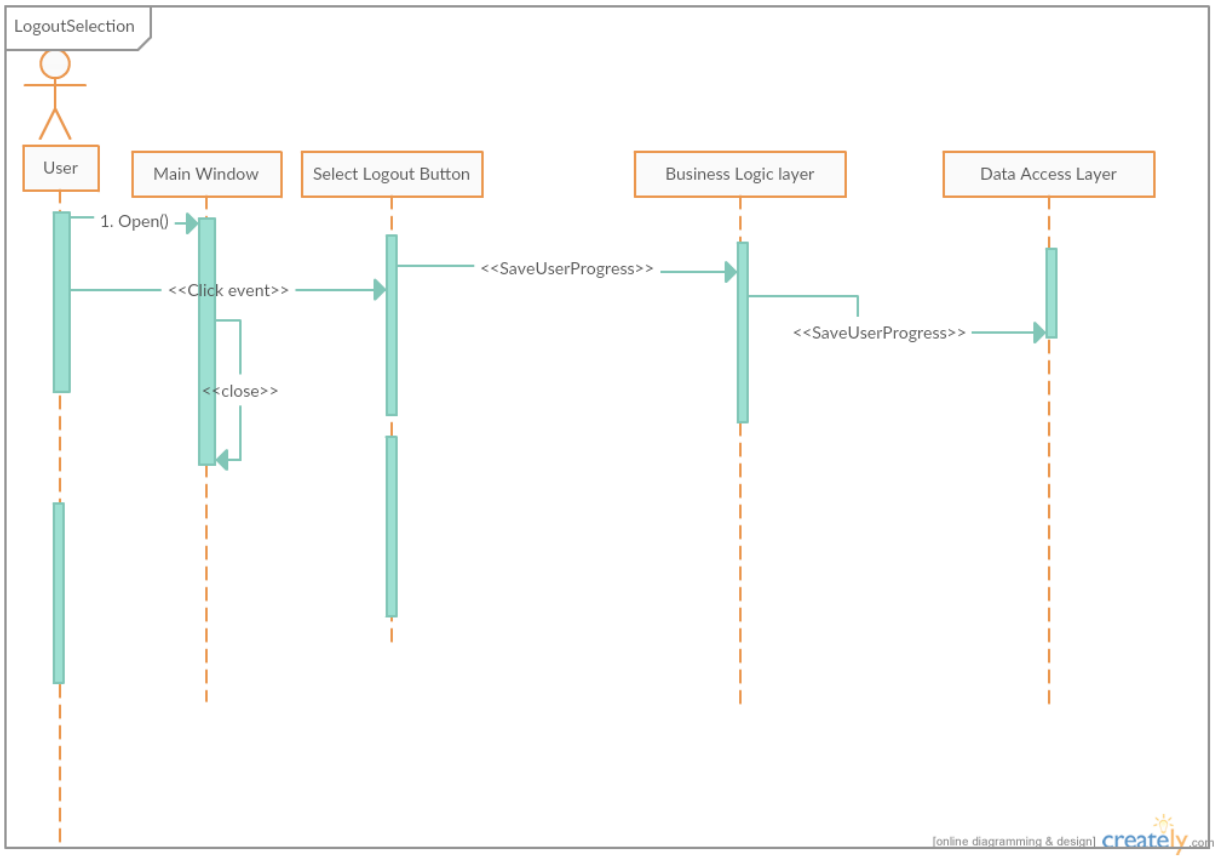


Figure Logout from the application

23d Subsystem Decomposition

The system decomposition has been divided into two separate parts. The first part is the product organization part and product design. The product organization part consists of backend setup and design of the product features. It is summarized in the component diagram below:

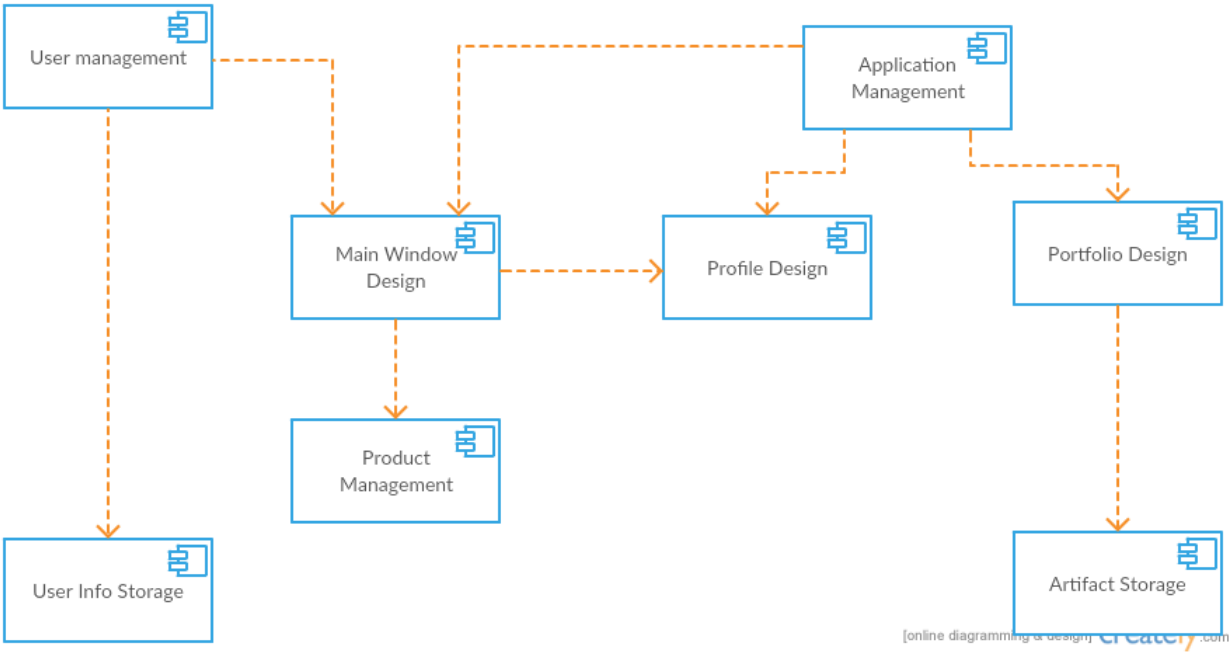


Figure Product Organization System

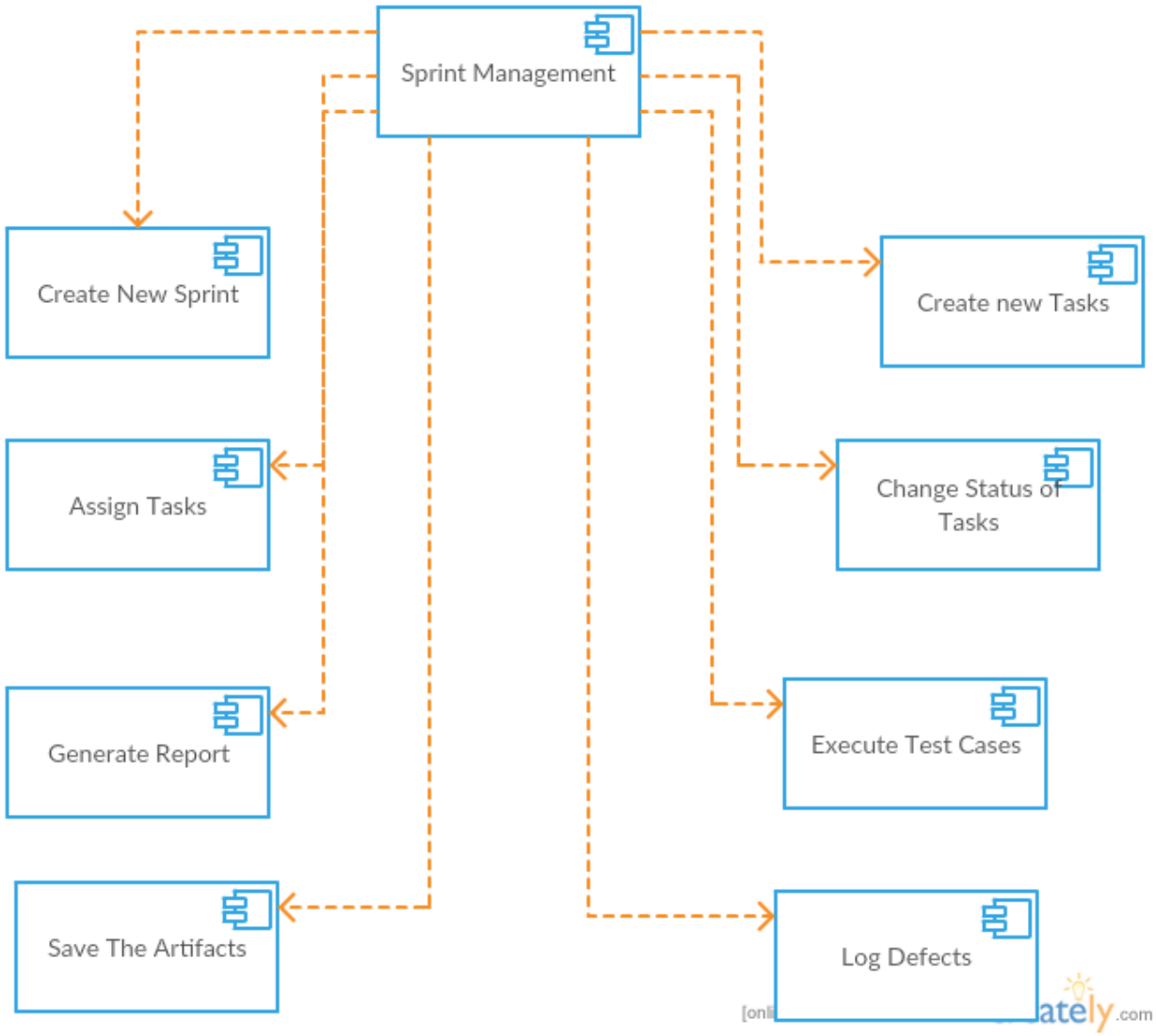


Figure Product Design Subsystem

23e Hardware / software mapping

The product is built upon a web server. The server will be responsible for storing the information of the user and the storage of the artifact repository.

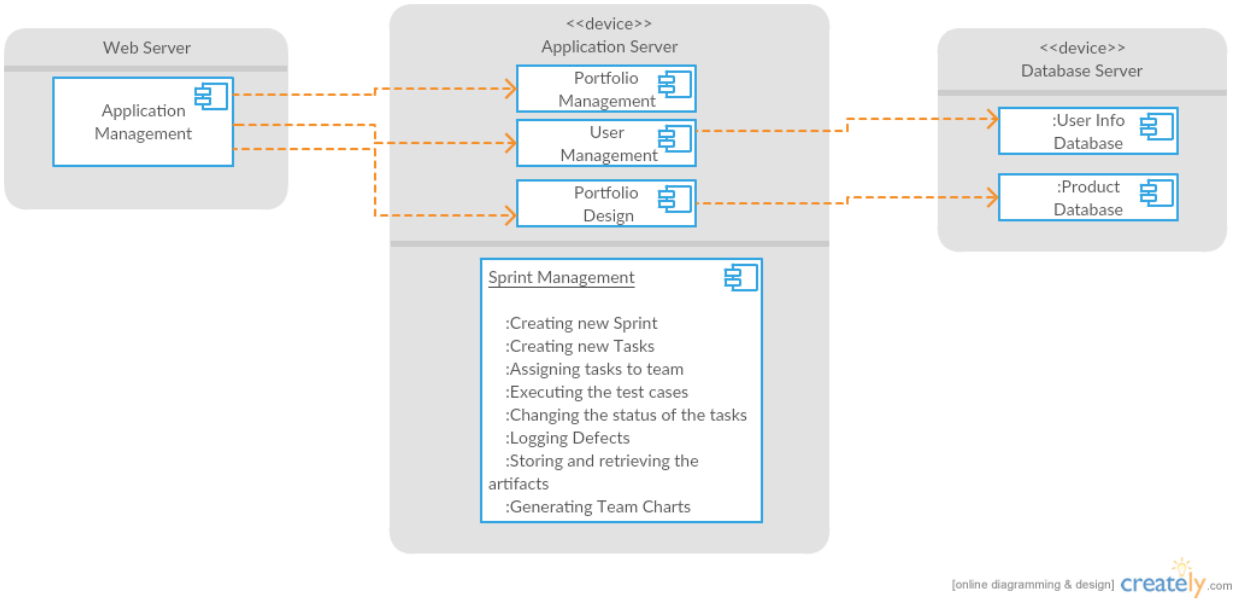


Figure Deployment Diagram

23f Data Dictionary

Successful Login:

When the user is successfully able to login to the system.

Sprint:

Beginning of the sprint cycle for the current project

Task:

Each sprint has multiple tasks and are assigned to different team members to work upon.

TestCases:

Each task can be subdivided into multiple user stories and from thereon to multiple test cases. These test cases are created by the user.

Scheduling of Task:

To divide or distribute the task amongst different team members

Defect Creation:

To create defect for each failed test case and providing steps to repro the defect.

Triaging of Defect:

To distribute the defect amongst the team members and work upon it checking the severity and priority of the defect.

Team Meetings:

The different team meetings the user will be participant of throughout the sprint.

BurnDown Report:

The activity or effort over time report of either each individual or the entire team for the entire sprint.

23g Persistent Data management

The persistent data includes the login information of the users that is entered into the database once the user signs up for the product. This persistent data is stored in the database. The database that will be used is a relational database. The database will be kept at a remote location and will only be accessible to game administrators.

Another persistent data that exists for this game is the career report of the users, which includes the total number of tasks completed by the user, total number of test cases created, executed and number of defects logged and validated.

23i Global software control

The control flow that will be selected for Ozil will be the event-driven control flow paradigm. Once the user requests some operation by clicking on a button, an event will become available and it will be dispatched to an appropriate object, based on the information associated with the event. This can include clicking on the "Login" button. Once the user clicks on the "Login" button, an event will be fired and an appropriate object will be dispatched which will result in a Login form appearing to the screen. This will be the case for almost all of the features of the product. Once the user clicks something, an appropriate event will be fired, resulting in a new window appearing or a new action taking place.

24 Subsystem services

- **Application Management**: Application management consists of maintaining the application and providing latest updates and prompting the user when an update is available so that the users have latest version of the product with them.

- **User Management**: The user management consists of keeping all the features of the product up to date for the users. The new user must be able to register for the product and the returning users must be able to login by supplying their username and password.

- **Main Window Design**: The welcome window shall contain the sign-in, the login and the exit buttons for the users.

- **Profile Design**: The users shall be given to view their profile as well as access their portfolio after they login or signup.

- **Portfolio Design**: The portfolio shall be defined in a way that allows user to view total effort he has put into the current sprint.

- **Portfolio Management**: The portfolio should consist of options like number of tasks taken for the current sprint, number of test cases created, executed, defects logged and validated and how many team meetings attended, quality of the defects found etc.

26 Object Design

26a Object Design trade-offs

- Changes vs Modification: Reusability is not the main concern for designing this project but open to changes and close to modifications. Therefore the object design and design pattern embrace to the change in this project. Any changes in the project will not force too much modification to another class or layer in the multi-tier design.
- Functionality vs Usability: It is very important to have wide range of users for this product. Therefore, the product will have plain usage. The system should not be too complex to use. It means the functionality of the system will be basic.
- Space vs Speed: Space is notably inexpensive and as such is more expandable than speed. It is more important that the system is responsive and doesn't take too much time to perform internal operation. Therefore the design of this project emphasis on speed.

26b Interface Documentation guidelines

Identifier Type	Rules for Naming	Example
Packages	The prefix of a unique package name is desired and will contain all lower case but the first letter.	Package Ozil;
Classes	Class name should be nouns in mixed camel case. The first letter should be upper case follow by every noun to start with first letter upper case.	Class User;

Interfaces	Interface naming convention should adhere the same naming convention from classes.	Interface Report;
Methods	Methods should having verb naming convention, so the receiver of the method has clear understanding for the purpose of the method. Naming verb should not be ambiguous.	createTask();
Class member field or variables.	All member fields and variables will start with lower case letter but global variables. Camel case convention should also be followed in naming conventions for variables. Common and ambiguous names i.e. x, y, z should be avoided at all cost.	String firstName;
Constants	All constant variables local and/or global should be all upper case separated by underscore '_'.	Const long MAX_USER_CONNECTION = 15;

26c Packages

The project will contain one package name across the system. Developers' team are at freedom to choose meaningful project + company they work for as part of their package signature.

26d Class Interfaces

- User
- Sprint
- Task
- TestCase
- Defect
- Status
- Priority
- Report

- Meeting
- Deployment

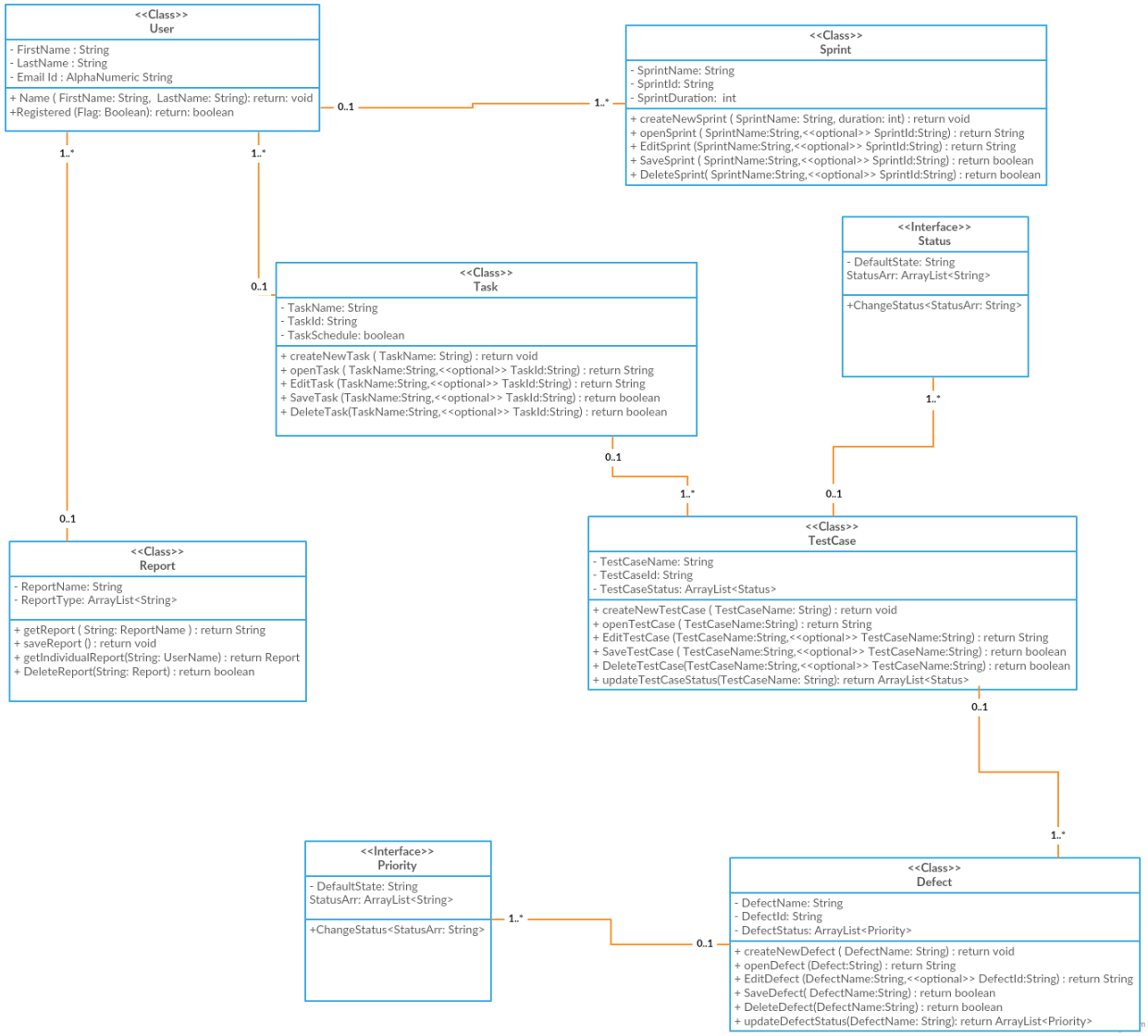


Figure Class Diagram

IV Test Plans

27 Features to be tested / not to be tested

Ozil, a CASE Tool must be thoroughly tested to ensure that it operates at a high standard. The following is a list of general features that will be fully tested before the final release.

1. New user sign up.

2. Returning user login.

3. User creates user story.

4. User creates task for the user stories.

5. User assigns the task to the different resources in the team.

6. User moves the task across various stages.

7. User logs in as manager and checks for the various report.

8. User logs in the repository and checks out the execution sheet.

9. User logs in the repository and checks in the execution sheet.

10. User should confirm that the execution changes made across the team is shown on the execution sheet.

11. The defects logged in by different users should be mapped to the failed test cases if any.

12. The proof (image/video) to the steps to repro for the defect should be properly updated or attached in the defect and on playback, in case of video, should be playable.

13. The execution sheet should be auto saved periodically and option to create different versions should be present. On opening the appropriate version of the file, appropriate file should be opened.

14. The user should check that if the internet connection goes down, the execution sheet does not close. So that the user can save his work locally.

15. in case if the site is under maintenance, the user should get appropriate error message and not 404 page.

16. on clicking the different tabs and different links, appropriate pages should be opened.

17. UI, font styling and spacing should be as per the requirements.

Features not to be tested are those that reside outside of our control, namely to do with the archiving of the sheets in the database or concerned with the defects that are closed and very old.

28 Pass/Fail Criteria

A test will be considered a pass if and only if the actual results of the test match the expected results specified in the associated test case.

Any deviation from the expected results will be considered a failure of that test.

A feature or method will pass when it has been subjected to all associated test cases and passes at least 99 percent of the time. Anything less will call for debugging of the associated code.

29 Approach

The testing approach for this project will be efficient. Testing will occur during all stages of development in parallel with other development tasks.

All developers are expected to perform white box path testing and unit testing, and generate associated reports, for any code they submit to the project.

A fellow developer must then inspect submitted code.

Once all code associated with a certain feature has gone through these first steps, integration testing may begin for the future.

Then the features will be black box tested by the associated test cases below.

As features are completed they will then also be run through integration testing to insure they work together.

30 Suspension and resumption

If at any point in the schedule any part of the test code does not pass the fit criteria, the development team's concern developer must be notified immediately.

If the developer is not present, the manager of the development team must be notified. No buggy part of the code shall ever be accepted. Developer should be informed regarding the test case and detail report of pass and test cases along with testing environment. The feature in question must be suspended with immediate action until the development team fixes the bug and code is re-tested.

31 Testing materials (hardware / software requirements)

Hardware:

- Dell PowerEdge Servers
- HP Server DL380p and ML350p
- Microsoft Surface Pro 3 and 4
- Apple MacBook Pro

- Amazon AWS Servers
- Microsoft Azure Cloud Servers

Software:
- OS-X Maverick
- OS-X Yosemite
- OS-X El Capitan
- Windows 7 Pro
- Windows 8.1 Pro
- Windows 10 Pro
- Ubuntu 14.04 LTS
- Linux Mint 17.2 Rafaela

32 Test cases

Test Name	Entry Condition	Flow of Events	Exit Condition
registerUser_Complete	User is on the home screen and is a new user	1. User clicks sign up. 2. User fills in all required fields and clicks done.	User's information is added to the user database and the user is brought to their portfolio page.
registerUser_Incomplete	User is on the home screen and is a new user	1. User clicks sign up. 2. User leaves one or more required fields blank and clicks done.	A dialogue highlighting the missing information is displayed. Info is not added to the database.
loginAccount_Correct	User is on the home screen and has an account.	1. User clicks Login. 2. User enters a valid username and password.	The user is logged in and brought to their portfolio page.

loginAccount_Incorrect	User is on the home screen	1. User clicks Login. 2. User enters invalid username and password.	The user is prompted that they have entered an invalid username or password

userStoryCreation	User has successfully logged in the application	1. User clicks on create New User Story link 2. User enters the name of the user story and clicks on Save button	The user story should be saved in the system and should be displayed under the User Stories tab.
checkUserStory	User has successfully logged in the application	User clicks on one of the user story that has been created by him or team member.	The user story should be opened up and should display the details of the user story.
taskCreation	User has successfully logged in the application and has created user story.	1. User clicks on create New Task link 2. User enters the name of the task and clicks on Save button	The new Task should be created for the user and he should be able to see it on the My Task screen under the user story section
taskScheduling	User has successfully logged in the application and has created user story.	1. User creates new task for the user story. 2. User assigns the task to any of the team member	The task should be assigned to the respective team member and should show up new task for that team member

taskReScheduling	1.User has successfully logged in the application and has created user story. 2. User has assigned task to his team member.	1. User reassigns the task to any of the team member other than the user he had previously assigned to.	1. The task should be assigned to the respective team member and should show up new task for that team member. 2. The task should be removed from the task list of the previous user.
taskTransition	User has already created user stories and have assigned certain tasks in the user stories to respective team members.	1. User moves the task from “to do” to “in progress”	1. The task card should be moved to “in progress” state 2. The task status should be appropriately updated to the new state. 3. The same status should be reported in the burndown chart should the manager opens it to have a view.

taskReTransition	1. User has already created user stories and have assigned certain tasks in the user stories to respective team members. 2. User has moved the task card to new state.	1. User moves the task from “complete” to “in progress”	1. The task card should be moved back to “in progress” state 2. The task status should be appropriately updated to the new state. 3. The same status should be reported in the burndown chart should the manager opens it to have a view.
testcaseCreation	User has got the task to develop the test cases of particular PBI	1. User creates new Execution sheet in the repository. 2. User opens the sheet in share mode. 3. User writes down the test cases for the user story. 4. User saves the sheet.	The saved sheet should be present in the repository along with the changes made to the sheet.
testcaseExecution	User has got the task to execute the test case of particular PBI	1. User executes the test cases and appropriately update the status of the test cases 2. User saves the changes.	The sheet should be saved along with the execution status in the repository.

testcaseSync	1.User has opened the execution sheet in share mode. 2. Multiple users are working on the same sheet.	1. The user clicks on save button.	The user should be able to see the work of all the people who have worked on the sheet.
defectCreation	1. User has executed the test cases and has marked some of the test cases as fail	1. The user should click on create new defect for the test case that has failed. 2. The user should be presented with log new defect screen. 3. The user enters the steps to repro the defect and attaches the screen shot and video to repro the defect. 4. The user assigns the defect to the developer after assigning it severity and priority. 5. The user submits the defect.	New defect id for the defect should be generated. The defect should be assigned to the developer.

defectTriage	1. User has already logged the defect and is the defect has been fixed by the developer and is sent back to the user	1. User enters his comments in the validation comments box. 2. User saves his comments. 3. User changes the status of the defect from open to fix validated(environment)->fixed->closed	The defect status along with comments should be appropriately updated.
managerReport	The manager has logged in the system.	1.The manager has clicked on the report tab. 2. The manager clicks on the burndown report tabs.	The burndown tab should open and show the efforts made by each resource in the current sprint
managerReport	The manager has logged in the system.	1.The manager has clicked on the report tab. 2. The manager clicks on the burnup report tabs.	The burnup tab should open and show the efforts made by each resource in the current sprint
videosTab	The user has logged in the system	1. The user clicks on the video tab. 2. The user selects one of the video from the menu and clicks on the video link.	The video should start playing
toolTip	The user has logged in the system	1. The user hovers the mouse on any one of the icon.	The tool tip for the icon should be displayed to the user

siteMaintenanceMessage	The site is down due to maintenance work	The user tries to log into the system	The user should get proper error message that the site is down for maintenance.

33 Testing schedule

Path Testing	Throughout Development
Inspection	Throughout Development
Unit Testing	Duration :1 Week, Begins: 3 Weeks before each release
Integration Testing	Duration :1 Week, Begins: 2 Weeks before each release
Usability Testing	Duration : 3 Days, Begins: 3 Weeks before each release
Integration Testing 2	Duration :1 Week, Begins: 1 Weeks before each release

V Project Issues

34 Open Issues

We are continuing to search for other viable sources that we can add more automated services and better git handling as well as something like voice to text converter. Currently, we have decided to obtain the information from the following products: GIT, ICESCRUM, TFS, QC, SalesForce, Trello, Slack, etc.

35 Off-the-Shelf Solutions

35a Ready-Made Products

Since ready-made products provide lower costs, minimal problems and unparalleled technical support, it would be wise to consider going that route for some software of hardware products during the development of this game. One of the products that can be taken from the market are the type of databases that will be used for this product. Two databases are needed, one for the user information and one for the game information. Microsoft SQL Server, MySQL and SQLite are among the few

candidates for the databases that can be used off the market. It will be the decision of the development team to decide which product to eventually use.

We are also comparing our product with : GIT, ICESCRUM, TFS, QC, SalesForce, Trello, Slack, etc.

35b Reusable Components

Instead of having to recreate hypothetical data and hypothetical historic information, as mentioned before, we will be using real historic information obtained ideally from the Products mentioned before. This will be a major help for the development team, as they can place more of their focus on the design, instead of having to worry about creating and modifying data throughout.

35c Products That Can Be Copied

As it turns out, there are quite a few other products on the market that are similar to the Ozil product. One such product is the ICESCRUM. However, it is very basic and does not provide the features we will provide.

36 New Problems

36a Effects on the Current Environment

Since this is a new desktop application, and as such, the current environment will depend on the development team and how they choose to develop this product.

36b Effects on the Installed Systems

The product will have to conform to the current popular operating systems, Windows, Linux, and Macintosh. Although it is up to the programming team, we suggest the usage of Java programming language due to its popularity and portability.

36c Potential User Problems

We hope to design the software to be intuitive and easy to use.

36d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The software is limited by the individual users' hardware. However, since most users in today's age have computers running on Intel Core i5 or equivalent processors as a minimum, and the product is not a graphics intensive game, then the great majority of users should be able to run the product with no limitations.

36e Follow-Up Problems

If the average number of logged in users exceed 150,000, then there may need to be expansions done to the user and product database in order to house a larger number of users

and run a larger number of application. This could also result in increased server traffic which will require upgrading the server parts to more powerful hardware.

37 Tasks

37a Project Planning

Software Development Life (SDLC) is used in the development of the software. It is a structure imposed on the development of a software product. There are several models for such life cycle, each describing approaches to a variety of tasks or activities that takes place during the process. Some people consider a life-cycle model a more general term and software development process a more specific term. For example there are many specific software development processes that 'fit' the spiral life-cycle model. An international standard for software life cycle processes aims to be the standard that defines all the tasks required for developing and maintaining software, details of the lifecycle and approach that will be used to deliver the product. The new product will be developed using SDLC. However some circumstances are unique to a particular product and will necessitate changes to the life cycle. While these considerations are not product requirements, they are needed if the product is to be successfully developed. Planning of the Development Phases

37b Planning of the Development Phases

The development of the Ozil is planned in the following ways. First would be the requirements phase in which functional and non-functional requirements would be gathered followed by the analysis phase and design phase. In implementation the coding would be done. It must be taken care that product should be able to deploy most of the Functional Requirements.

38 Migration to the New Product

38a Requirements for Migration to the New Product

On availability the users who will move over to the new application will not face a lot of migration problems in terms of operating systems. The product will be compatible with windows, Mac OSX and Linux, hence users that already have these devices will be able to run the application efficiently.

One of the main constrains will be that the users will face a roadblock because of the additional hardware that is required and the users with old systems will need to upgrade their system.

We suggest some minimum requirements for smooth operation of the application:

- The minimum RAM required to run the application is at least 2GB.

- At least 50Mbps of internet speed for the smooth transition between different pages.

- The minimum screen resolution for the system should be 800 x 600 for better display of graphics.

38b Data That Has to Be Modified or Translated for the New System

There are no such specific data transitional requirements for implementation of the product as the product will be in accordance of the existing data resources available on the application. There would be no importing of data from any other previous applications.

However there is an exception in the case of new product updates, the application might require specific data based updates but this will be an automated process and developers creating updates will be required to ensure consistency from internal testing before the updated release is available in the market.

39 Risks

Ozil will involve risks—namely, the risk of inaccurate metrics, inadequate measurement, inaccurate cost estimating, low quality and low productivity of the software. Risk is only a bad thing if the risks are ignored and they become problems. Risk management entails assessing which risks are most likely to apply to the project, deciding a course of action if they become problems, and monitoring projects to give early warnings of risks becoming problems. Following points will provide a transitory but comprehensive illustration to such causes.

• Inaccurate metrics: Inaccurate information regarding user, sprint, user details, execution details, defect details, user charts, and execution summary can result in user losing their confidence in the software. We have to constantly remember the fact the core reason for this project is

developing a tool that can be used to track down each milestone of the user while developing product in SDLC.

• Inadequate measurement: It is important to understand the target audience for this product. If the product is repeatedly showing incorrect output for the users, text execution and defect details as well as sprint planning milestone, probability is high that they will not want to use this product again in the future.
It is important to understand the target audience and also deploy algorithms within software product to encounter such issue.

• Inaccurate cost estimating: Providing an inaccurate cost to the eventual client regardless it is high or low can deeply harm the project before even it begins. It is important that the current market price analysis should be done to get better estimation to provide more accurate quote to the customer, who is willing to pay for this project.

• Low quality: Low quality software can lead to bad end user experience and eventually bad reviews and disaster for the project to grow more customer base. It is important that this project provide ease of use user interface, means for the customer to provide feedback regarding their experience and opt in option for the user to provide stack dump on every crash to determine the cause. Such issues must be taken seriously and provide solution as soon as possible.

•Low productivity: It is also important to provide something new i.e. feature or change in user interface design to the customer. Humans tend to get bored from using one thing over and over, therefore it is important to bring high productivity for this project and add something fresh and thrilling for the end users for every new release.

40 Costs

Ozil will necessitate the following cost factors and it is important to remember this is not hard coded limit set for the project management to not to analysis other various factors since changes are well expected at the time of development of this project:
● Number of input and output flows on the work context
● Number of business events
● Number of product use cases
● Number of functional requirements
● Number of nonfunctional requirements
● Number of requirements constraints
● Number of function points

41 Waiting Room

Other requirements that we would like to be implemented in the future releases include:

- Include automated phone bots for helping the users select phone bots for successfully identifying and mitigating their problems.

- Include miniature version of the team's performance in the current sprint on the run without having the manager or the super user to load the whole chart and check for the user's performance in details.

- A separate page for capturing different meetings through-out the project and differentiating between those meetings can be formed such as retrospective meetings, documenting down minutes of meeting and the topics discussed during that meeting.

- Different tools have to be downloaded like screen capture for capturing each screen and weaving through each of them to form a video for reproducing the steps to repro a defect. Instead, we could have an in-house tool for doing the same thing.

42 Ideas for Solutions

In order for the product to be more effective through-out the user base is to make the entire product more in-house and less reliable on different software and platform specific.
Like for defect logging, tracking, execution of test cases and their status updates generally use Microsoft products and that can be replaced by the tools and software from in-house.
A chat feature can also be included which enables user to share and discuss about various topics they come across.

43 Project Retrospective

This development project is being developed using the Waterfall Software Development Life Cycle. It was very helpful to divide the report into separate sections so that each section focused on one part of the report, such as requirements, design and testing report. The communication between the group always challenging at the beginning, but we were able to coordinate times to accommodate everyone's schedule. It is essential that the team coding this project also find proper ways of communication that work for everyone in the group early in the development stage. Team members should also openly think about any problem and brainstorm for the solution.

VI Glossary

The glossary defines terms that may not be familiar to all readers.

Error 404: If the server is down or any part of the webapp is unreachable, we are liable to get error 404.

GoTeam101: this is the by-default name of the sprint whenever the user does not enter sprint name and proceeds further. This is also the name given during the demonstration of the product.

VII References / Bibliography

This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the "Insert Citation" and "Bibliography" buttons in the "Citations & Bibliography" section under the "References" tab of MS Word. Creating new citations will not update this list unless you click on it and select "Update Field". You may need to reset the style for this paragraph to "normal" after updating.

[1] Christina Wallin. Software Development Lifecycle Models

[2] Young Hoon Kwak and Frank T. Anbari, Analyzing project management research: Perspectives from top management journals

[3] Pollyana Notargiacomo Mustaro & Rogério Rossi Project Management Principles Applied in Academic Research Projects

VIII Index

This section provides an index to the report. The sample below was generated using the "Mark Entry" and "Insert Index" items from the "Index" section on the "References" tab, and can be automatically updated by right clicking on the table below and selecting "Update Field". To remove marked entries from the document, toggle the display of hidden paragraph marks (the paragraph button on the "Home" tab), and remove the tags shown with XE in { curly braces. }