Software Requirements Specification

Scrum management system

Table of contents

Introduction

This projects emphasis an information system software engineering project concerned with handling a Scrum management system on an abstract level including the basic and main requirements needed to establish such system.

To facilitate communication, It’s decided to establish our software as a web application

The project in its design and implementation aims to reflect simplicity and agility concepts through preparing light weight , well emphasizing SRS and implementing the least needed features to have a working software.

We expect that the reader is familiar with Software engineering concepts specially agile, and popular illustration diagram.

purpose

The main purpose of the system is to provide the users with the essential tools to handle processes, updates, facilitate communication of Scrum managed project and bring all of those features together on a web application.

Glossary

Work	Definition
User	Regular user has a user name and password can sort or view or add task
Admin / Administrator	Can do what any regular user can do plus he can modify task or add back log.
Product backlog	A list which contains the system requirements
Sprint backlog	A sub list of system requirements that is created by the team for each sprint
Agile	Software development method that includes main features that helps in creating softwares faster and more efficiently
Scrum	A method the determines the details of implementations of and agile projects It's one of many other methods.
Scrum master	Team Leader also acts as normal member Considered as an essential part of a scrum management method
SRS	Software Requirements Specification, documents that contains the needed requirements by the customer.
Software life cycle	model represents all the activities required to make a software product transit through its life cycle phases.
Use Case	Diagram that consists of a series of actions that a user must initiate with the system to carry out some usefulwork and to achieve his/her goal.

Overall description

Scrum management system is a system that handle user stories and backlogs of the projects so first we have to types of user

1. Regular user
2. Administrator

each one has his own authorities and effect on the system and for more details you can see the use cases part, second part is the system itself in the home page we see a list of backlogs and we can add more backlogs depending on the system you develop , each backlog is a link to other list .
If you pressed on any backlog it will take you to another list which is the place of user story , this user story page contain a list of stories each story is a link to another page and the page contain more info like description and number of tasks in each story , we will explain all the details in next parts let's just focus on the headline and leave details later but first make sure to be understood the previous parts .
If user selected a user story , the tasks page will be appear , each task has green or red color, green means the task is completed and red means the task not finished yet and also the user can sort tasks but as we mentioned we won't go in details , but the user has option to mark a task to be finished if it's already finished we don't want any kind of cheating that will make your team angry and you may be get fired so please be honest . And the user has option to sort the tasks.

This is just a general description about the system and how it works but for more details just go to next parts and we will explain every think in details.
So we can conclude from all this description the following
The system consist of:

Actors

1. Members
2. Scrum master

Backlogs

1. Product backlog
2. Sprint backlog

Scenarios :
it includes:

1. Authorities and capabilities of different actors on different parts of the projects
2. Modifications made on data and flow of data

Functional Requirements

View Backlogs / Sprints / Tasks	
Brief description	The user views the tasks ,their status(completed or uncompleted) and the one responsible for them.
Actor	Members
Condition	The user is already registered and signed in.
Steps	1. Choose the desired project 2. A list of sprints will be displayed 3. Choose the desired sprint
Exceptions	None

Sort Tasks	
Brief description	The user views the tasks sorted according to their importance or their deadline date.
Actor	Members
Condition	The user is already registered and signed in.
Steps	1. Choose the desired project 2. A list of sprints will be displayed 3. Choose the desired sprint 4. Choose the desired sorting method from the combo box
Exceptions	None

Choose	
Brief description	The user can choose the desired task to work on.
Actor	Members
Condition	The user is already registered and signed in.
Steps	1. Choose the desired project 2. A list of sprints will be displayed 3. Choose the desired sprint 4. Press choose for a vacant task (colored in red)
Exceptions	Member can mark as completed when finished

Modify tasks state	
Brief description	The user modify the status of the tasks by applying for it and updating its status to complete.
Actor	Members
Condition	The user is already registered and signed in and applied for the task to be modified.
Steps	1. Choose the desired project 2. A list of sprints will be displayed 3. Choose the desired sprint 4. Mark the task as completed
Exceptions	The member can’t change the status of the task if he is not responsible for it.

Modify tasks state	
Brief description	The user modify the status of the tasks by applying for it and updating its status to complete.
Actor	Members
Condition	The user is already registered and signed in and applied for the task to be modified.
Steps	1. Choose the desired project 2. A list of sprints will be displayed 3. Choose the desired sprint 4. Mark the task as completed
Exceptions	The member can't change the status of the task if he is not responsible for it.

Assign tasks	
Brief description	The user could modify the responsibility state of a task.
Actor	Scrum Master
Condition	The user is already registered as scrum master and signed in, some member should have applied to the task.
Steps	1. Choose the desired project 2. A list of sprints will be displayed 3. Choose the desired sprint 4. Withdraw task
Exceptions	This could be done by modifying the task.

Modify tasks	
Brief description	The user could modify the task details.
Actor	Scrum Master
Condition	The user is already registered as scrum master and signed in. The task already exists.
Steps	1. View Tasks 2. Modify the task includes priority and deadline and description and the responsible member.
Exceptions	None

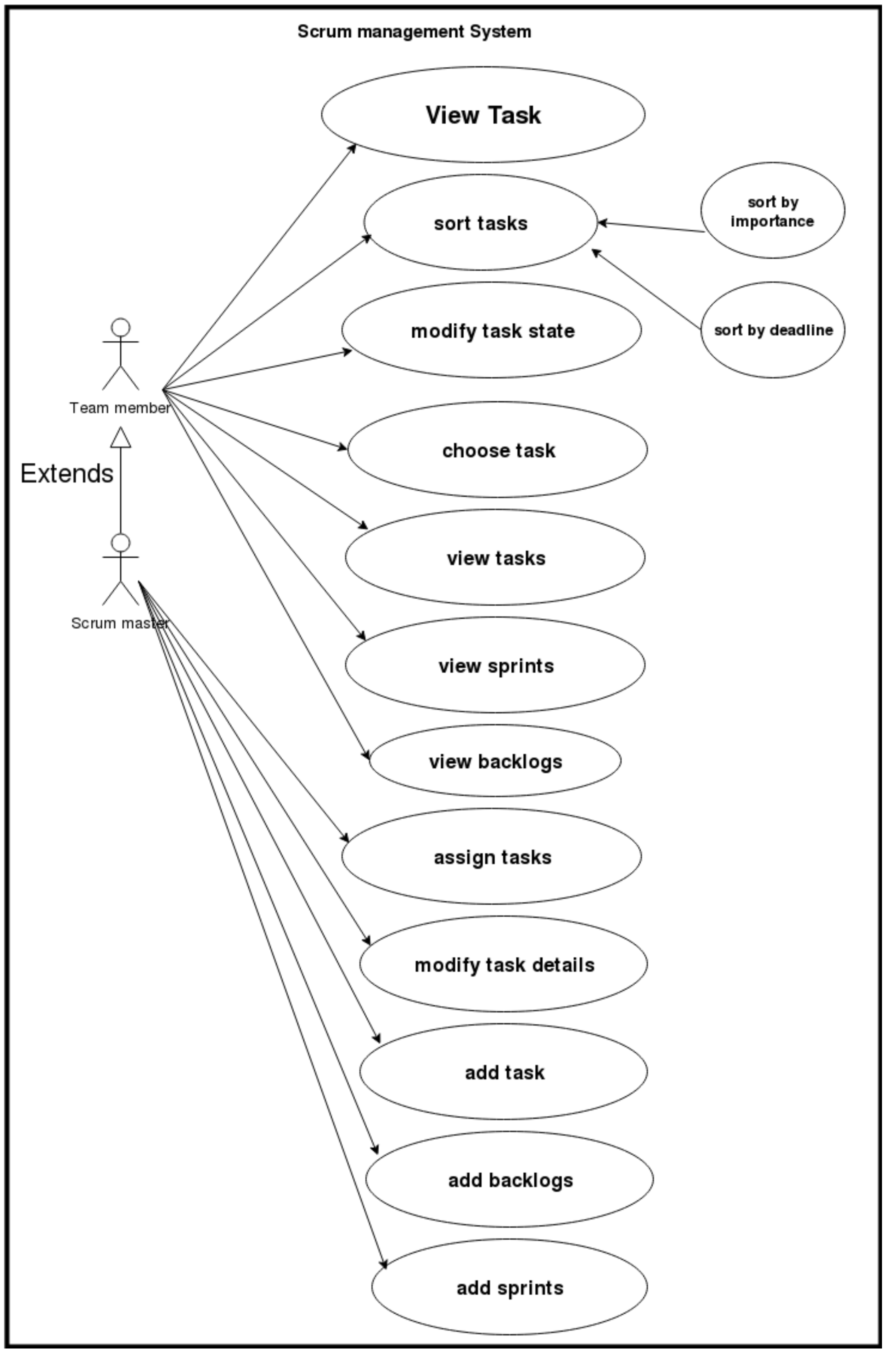
Add Backlogs / Sprints / Tasks	
Brief description	The could add projects(backlogs), or sprints within certain project or tasks within certain sprint
Actor	Scrum Master
Condition	The user is already registered as scrum master and signed in.
Steps	1. Add backlogs if desired 2. Choose desired backlog 3. Add a sprint if desired 4. Choose desired sprint 5. Add task if desired
Exceptions	This could be done by modifying the task.

User Characteristics

The user should be computer literate, familiar with agile scrum concepts and already confirmed on its role.

use case

This is the use cases which has two actors Team member and Scrum master scrum master can do what team member can do but he has more power than him he can assign or add tasks, and he can add sprints.
(note: sort tasks can be done either by importance or by deadline)

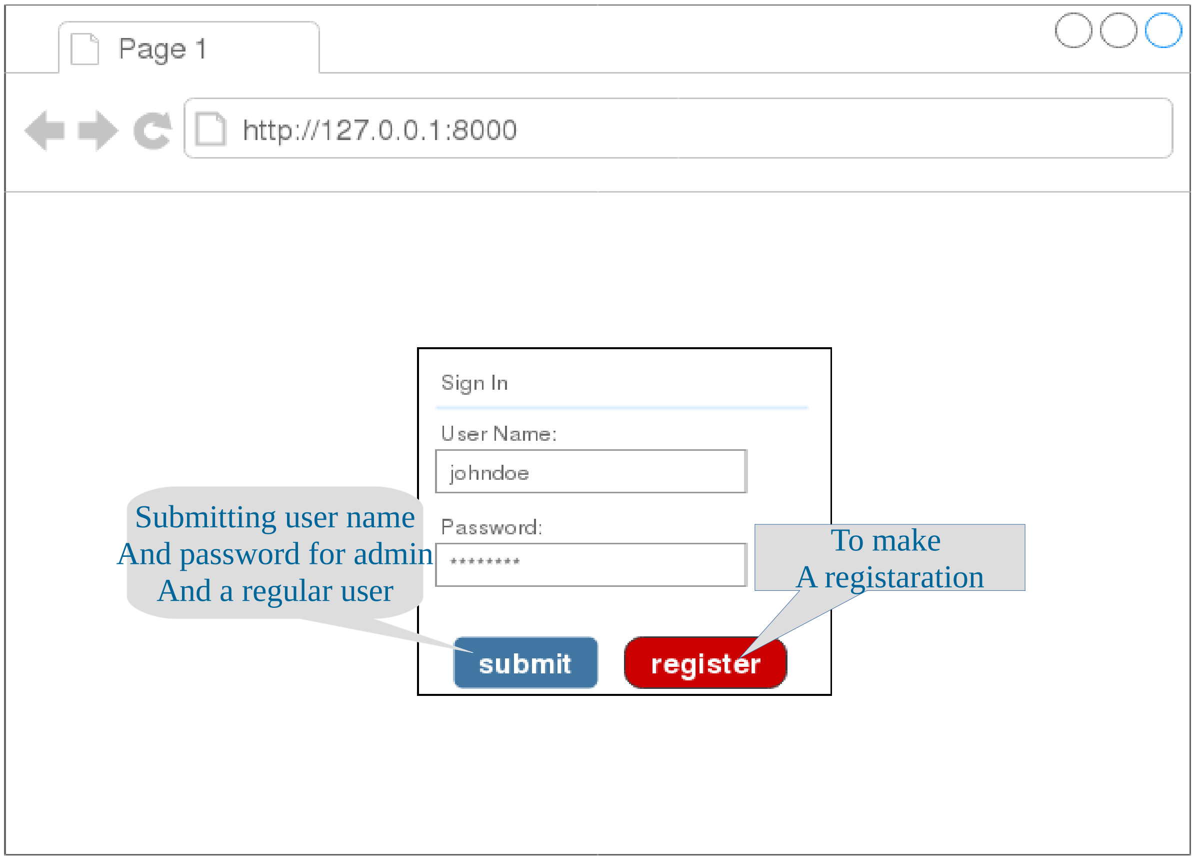


Wire frame

Define what needs to be done and also to have a clear picture of where we are going.
this is an over view of the system on how it should be look like it , it may or may not be as same as the system in gui but it has to be identical to the system in functions.
So this how our system will be look like starting from the starting page to every like in our system

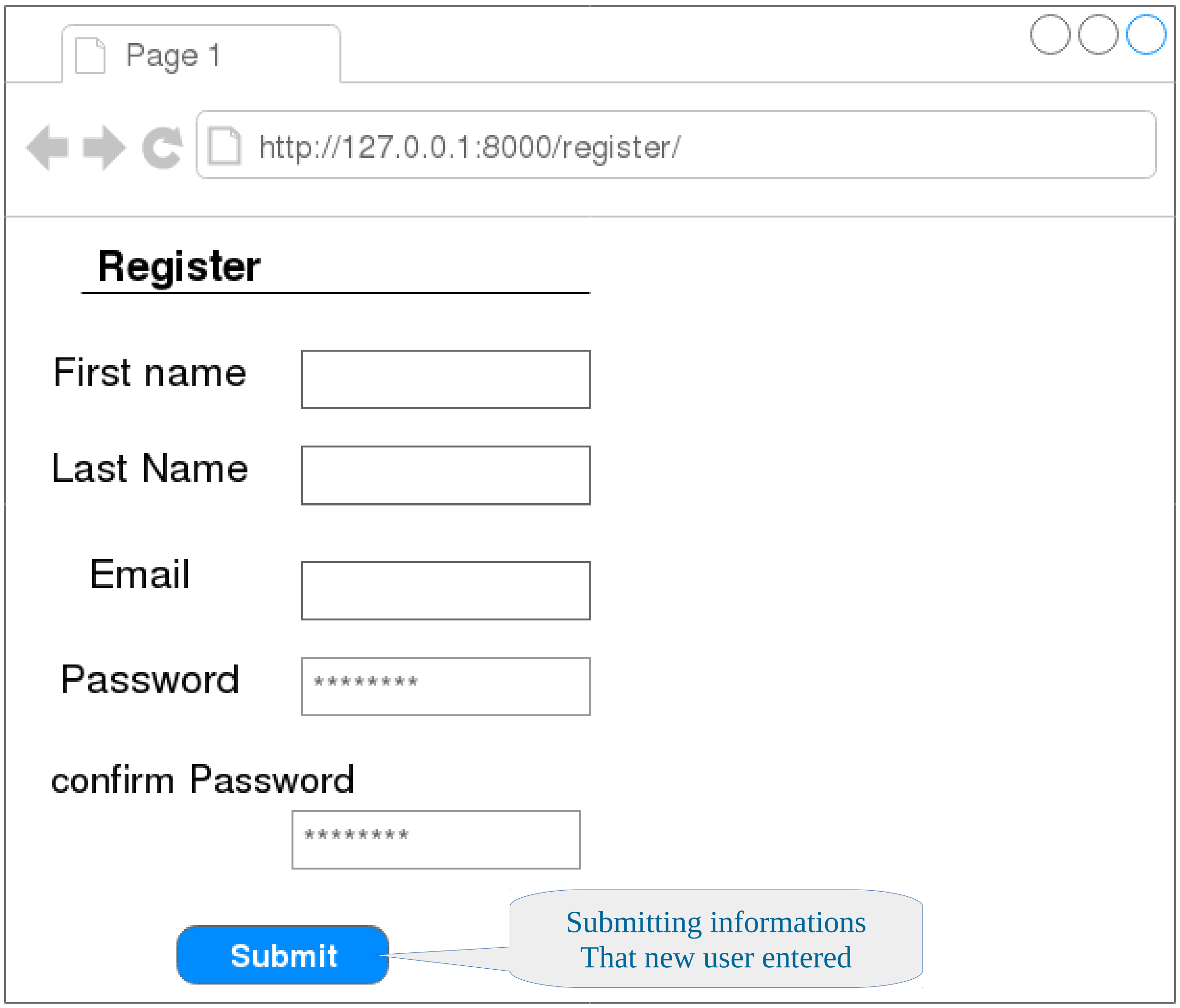
Login page

This is the starting page of our system the visitor has to sign in to see the home page of our system .We have two types of users either be a regular user or admin.



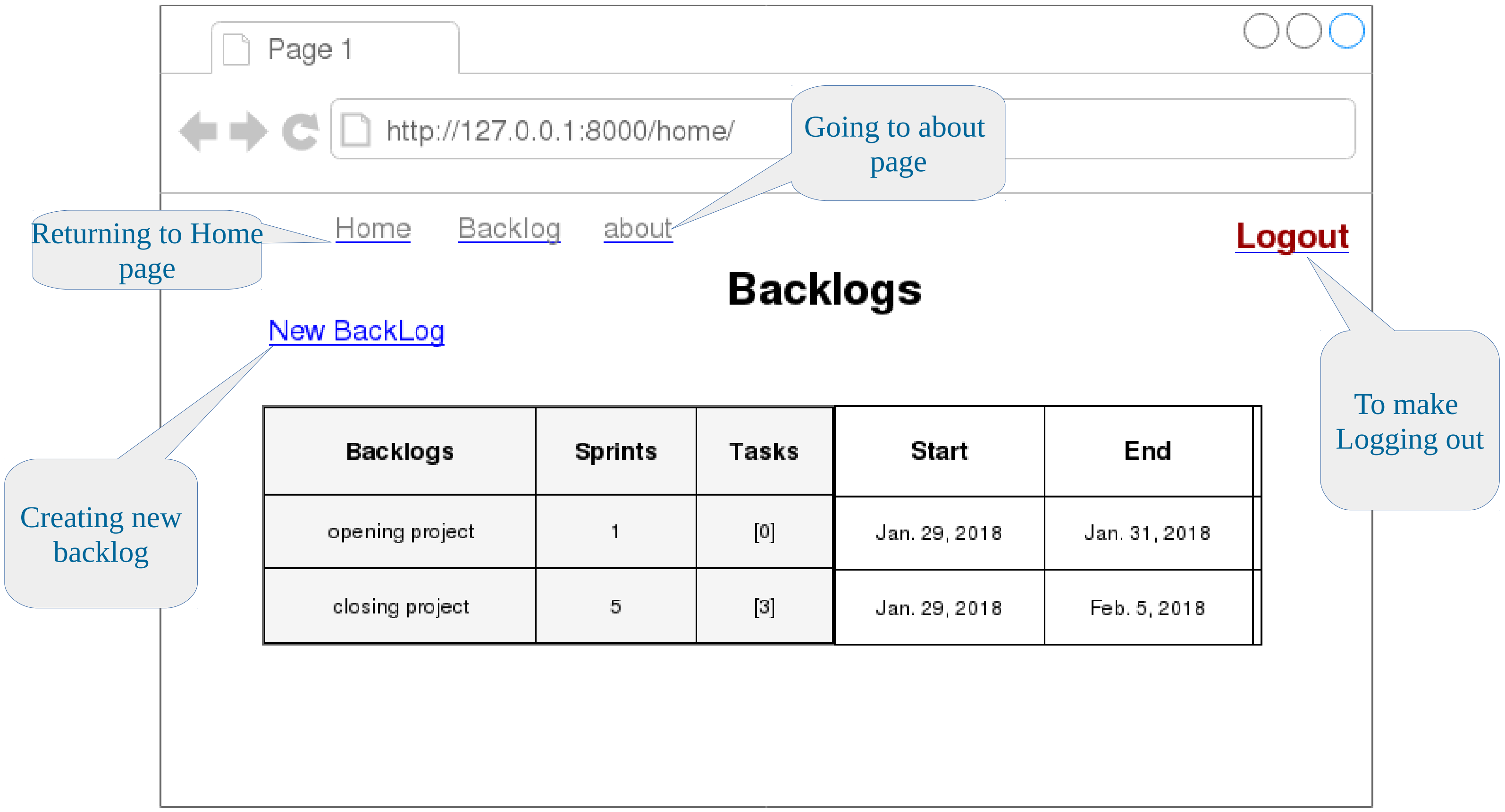
If the user chose register the following page will be disappeared.

The new user will insert his First and Last name and email and password.



Home page

After the user or admin login to our system he will see this page which consisting of list of backlogs , sprints , tasks , starting date and ending date.
he can create a backlog or open of this backlogs

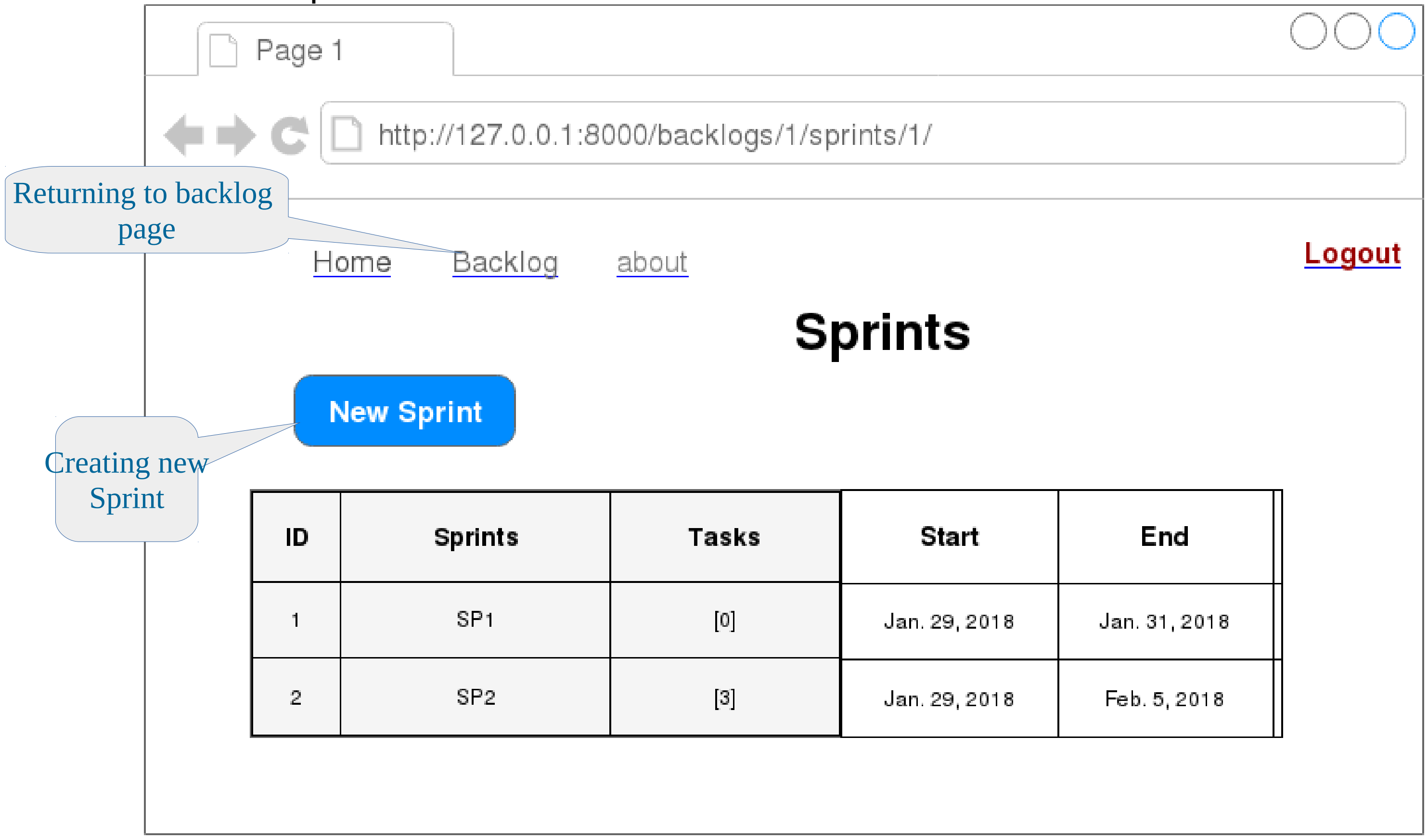


Backlogs

Backlogs	Sprints	Tasks	Start	End
opening project	1	[0]	Jan. 29, 2018	Jan. 31, 2018
closing project	5	[3]	Jan. 29, 2018	Feb. 5, 2018

Choose a backlogs

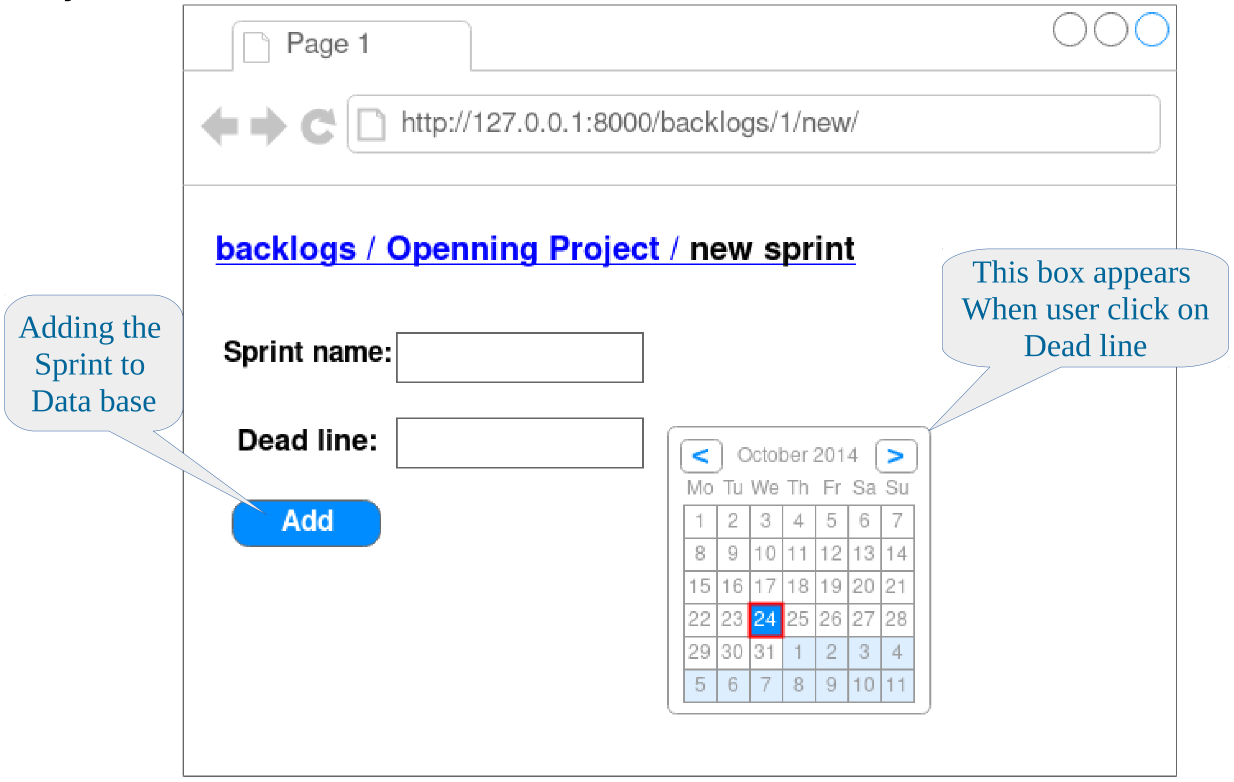
If the user chose opening Project he will see a list of sprints and IDs each sprint has starting and ending date.
He can choose either opening the sprint or make new one by pressing on New sprint button.


Page 1
http://127.0.0.1:8000/backlogs/1/sprints/1/
Returning to backlog page
Home
Backlog
about
Logout
Sprints
New Sprint
Creating new Sprint

ID	Sprints	Tasks	Start	End
1	SP1	[0]	Jan. 29, 2018	Jan. 31, 2018
2	SP2	[3]	Jan. 29, 2018	Feb. 5, 2018

New sprint

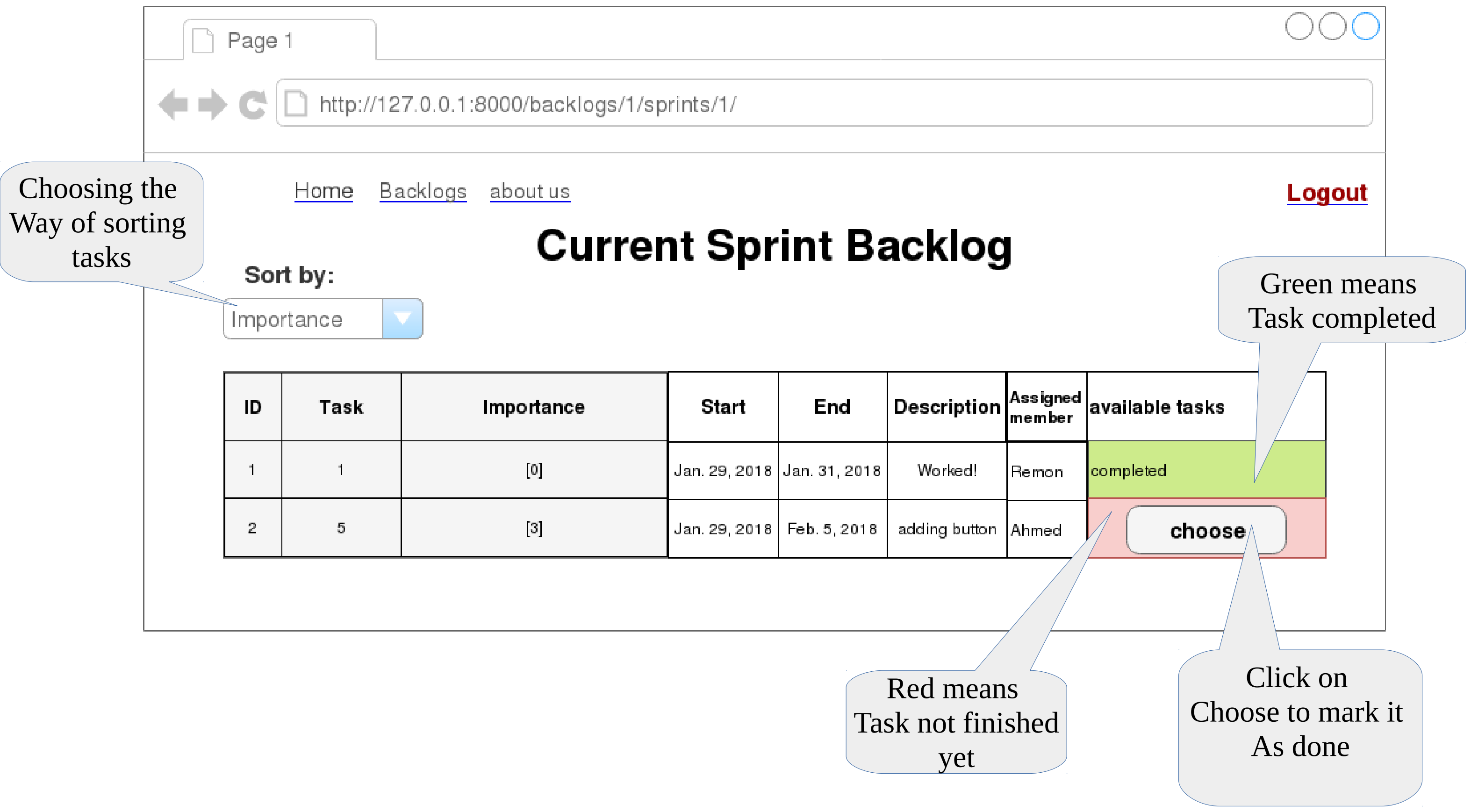
After the user choose new sprint he will has to fill the following requirements which consist of Sprint name and Dead line.
when the user click on dead line he will see a small of squares to choose the date from.
(note : user can't add new sprint unless previous one hasn't finished yet)


Page 1
http://127.0.0.1:8000/backlogs/1/new/
backlogs / Openning Project / new sprint
Sprint name:
Dead line:
Add
October 2014
Adding the Sprint to Data base
This box appears When user click on Dead line

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

5- choosing sp1

after choosing SP1 the user will see a list of tasks each task has it's own id plus two new things Description and status
Description: a brief description of the task
status : to determine the status of each task
available tasks : can choose either to make task is done and the color will be green like first task or leave it the same and the color will be red
sort by : the user has an option to sort tasks either by importance or which task is done.

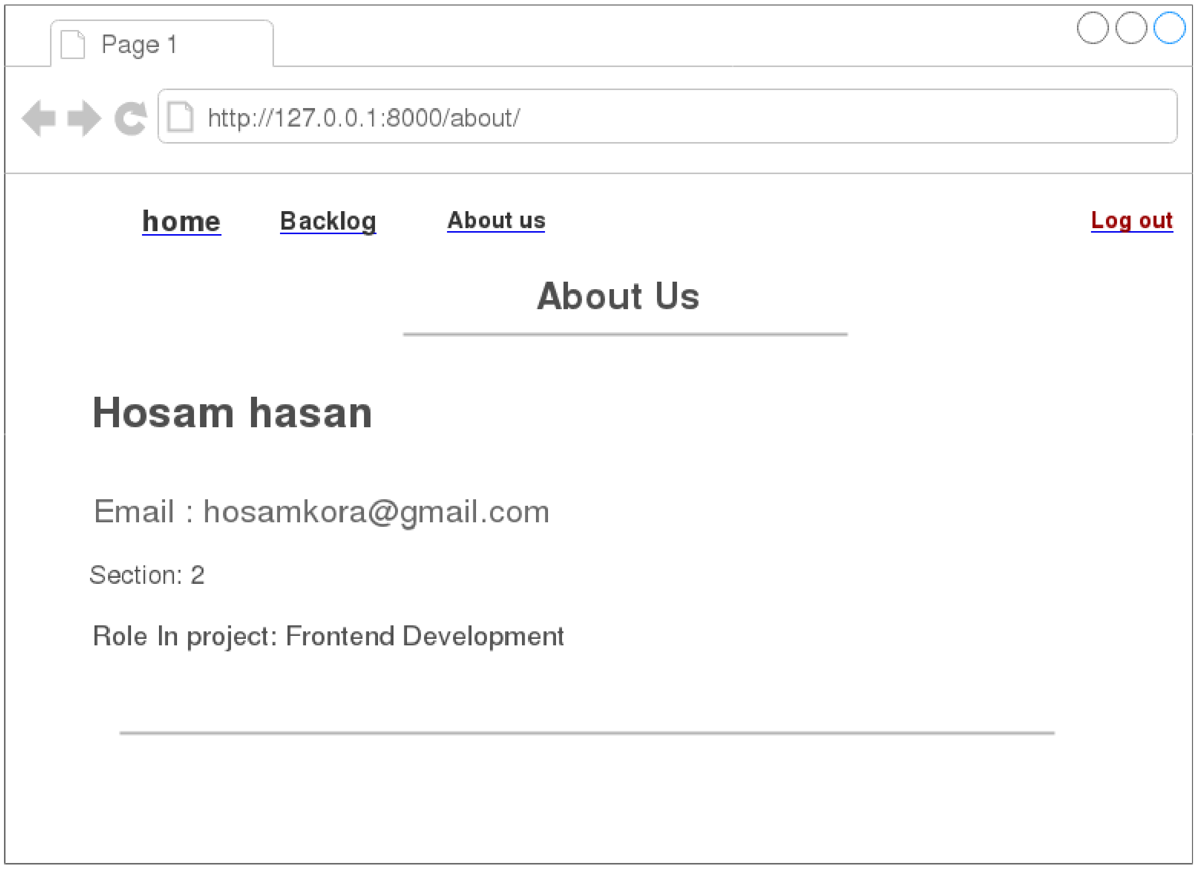

Page 1
http://127.0.0.1:8000/backlogs/1/sprints/1/
Home
Backlogs
about us
Logout
Sort by:
Importance
choose
Choosing the Way of sorting tasks
Green means Task completed
Red means Task not finished yet
Click on Choose to mark it As done

Current Sprint Backlog

ID	Task	Importance	Start	End	Description	Assigned member	available tasks
1	1	[0]	Jan. 29, 2018	Jan. 31, 2018	Worked!	Remon	completed
2	5	[3]	Jan. 29, 2018	Feb. 5, 2018	adding button	Ahmed	choose

About

Contain the team who worked on the system



About Us

Hosam hasan

Email : hosamkora@gmail.com

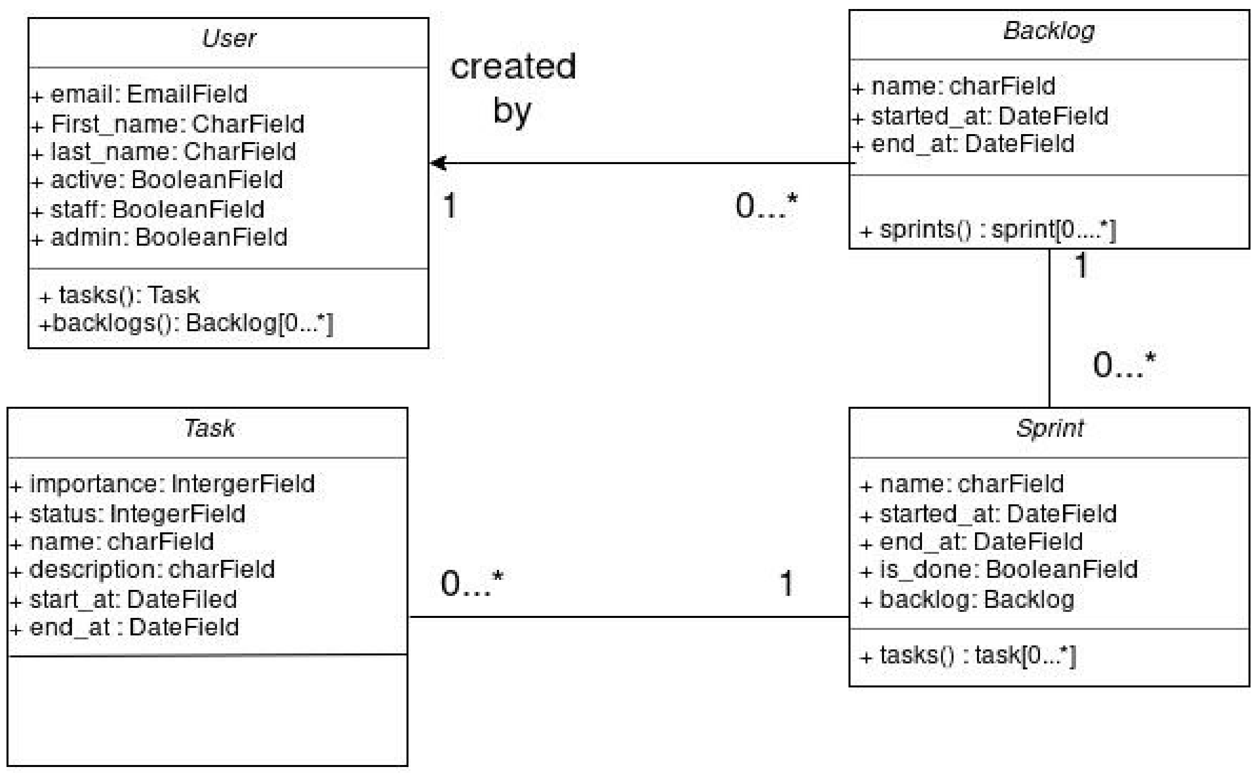
Section: 2

Role In project: Frontend Development

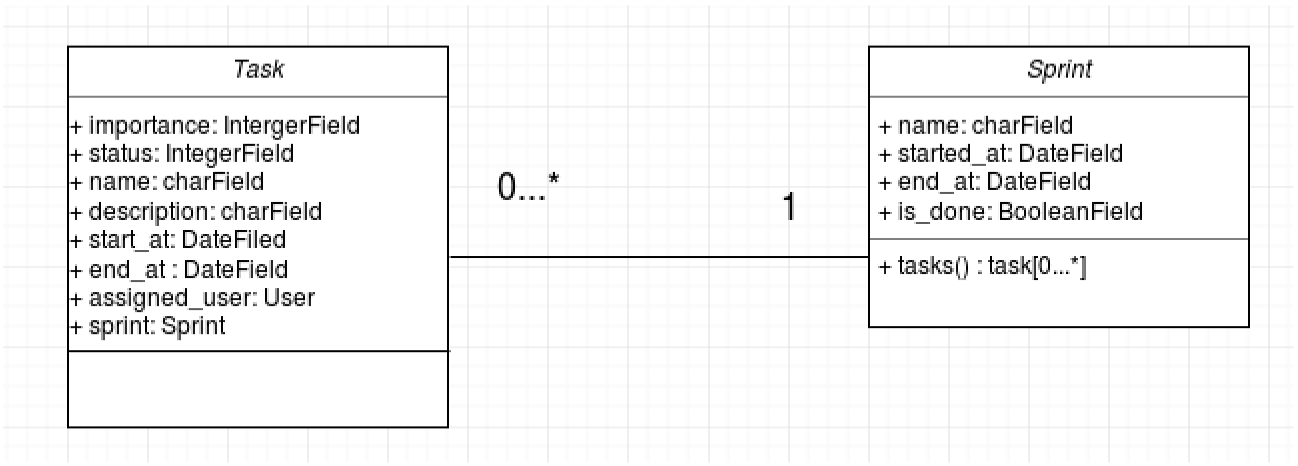
Class diagram

Our system has four classes User, Backlog, Task, Sprint
the user can create back log, and each backlog consist of sprint and task.
User can add a task or backlog as described in the class diagram , the arrow mention that backlog created by the user so that means each backlog is associated with the creator (user) and also we see [0....*]

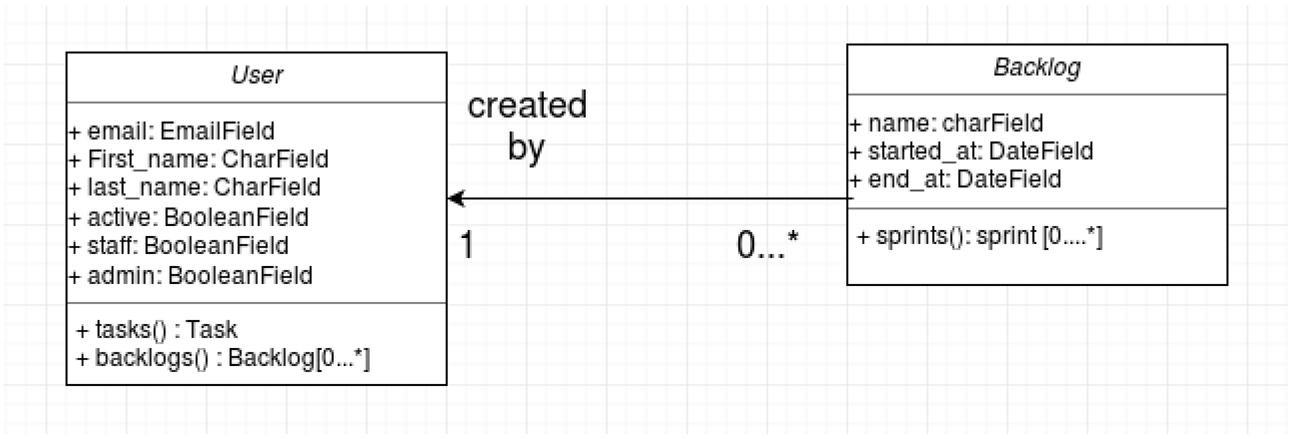
which means a user can create any number of backlogs, and also from the diagram backlog can has any number of Sprints.



- A task must be associated with exactly one (1) Sprint



- A user is associated with backlogs which he created



Non function requirements

Choosing how the data is stored and the most secure way of sorting all the data of the system, but we won't go deep in this section because django handle all this kind of feature, it handle how data will be stored and what kind of encryption this data will be encrypted.

Priorities and release plan

Our priority was building how our system will store and move between pages we didn't gave much attention to graphical user interface or to the documentation at fist we focused on the core of the system then first we started making the GUI and documentations in parallel, and for more details we created other file describing how we managed our system and how we divided our priorities and what should be finish first and what next and the amount of time for each task and how will we created risk analysis .
So in general we created the following in series:

- building the core parts of the system
- creating GUI and documentation in parallel
- testing the final product
- submit our work

Resources

- https://github.com/RemonIbrahimNashed/ScrumManagementSystem.git
- http://draw.io/
- https://simpleisbetterthancomplex.com/serie/
- http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf