Vision Document

for

BackTrack

Version 1.0

Team Kay

October 8 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Final Vision doc	06/12/19	Changed based on the final product	2

1. Business Requirements

1.1. Background

Our client, Digital Service Group, is a large firm with multiple different development teams. Currently, each development team has its own unique methods of managing their development cycle. For example, some of them are using a physical whiteboard to keep track of tasks. Managers find it increasingly difficult to oversee the progress of multiple projects without a centralized system. Developers also find it increasingly difficult to keep track of tasks that are important and vital for the project. Thus, our Client feels the need to build a custom standardized and centralized system for all development teams, to help them develop projects under the Scrum framework.

1.2. Business Opportunity

With this new system implemented, Development teams would be able to access the details of their current project online. This system would be extremely beneficial for Managers (Scrum Masters) who will be overseeing multiple different projects. This would help them examine the progress of their teams right from their mobile devices and PCs. This would save the manager a lot of time and help him/her to manage their projects efficiently. This system would also help manage their project tasks. Developers would be able to see the task that they have to complete in the current Sprint and at the same time understand the priority of each task. Hopefully, this would motivate developers to complete more essential tasks first and add further value to the project. A future extension of the system might be to track the activities of developers and make a better judgment during performance appraisals.

1.3. Vision Statement

BackTrack is a web application that is used to help development teams at Digital Service Group (DSG) to manage their development projects under the Scrum framework. Upon implementation of this system, managers would be able to oversee multiple different projects and have access to view the progress of Product Backlogs and ongoing Sprints online. Developers could also use the system to manage and take ownership of tasks.

Previously, Managers would have a harder time finding the product backlogs of teams since different teams might store/utilize them differently. Also, developers would have to keep track of tasks on their own and could hardly see what other developers are working on. With this system, we propose a further enhancement in the collaboration amongst development team members and thus revamp the efficiency and productivity of the team.

1.4. Business Assumptions and Dependencies

AS - 1: User is an authenticated developer / manager

AS - 2: User is aware of Sprint methodologies and understands the operating constraints of Scrum, and complies by them.

AS - 3: Users are limited to either developers or managers. They function and manage the project under their specific roles.

DE - 1: Our team will be using Django, a Python based open-source high-level web framework, for development of the server-side of our system.

DE - 2: Our team will be using ReactJS, a JavaScript based framework for making the frontend of our system.

DE - 3: The deployment and hosting of the system will use a cloud platform service called Heroku

2. Scope and Limitations

2.1. Major Features

FE-1: Scrum Master / Manager to be able to track multiple projects along with their Product Backlogs.

FE-2: Product Owner able to create, edit, and delete PBIs to Product Backlog.

FE-3: Developers able to create new sprint by selecting certain PBIs and defining tasks under each along with their effort hours. They will also be able to manage/edit current sprint.

FE-4 Developer will be able to select tasks they want to work on and take ownership.

FE-5: Everyone can view Product Backlog.

FE-6: Provide system access through the web through an authorisation and authentication page. Access would be defined through each user's role (Manager/Product Owner/ Developer)

FE-7: Product Owner is able to create a new project, invite a manager and other developers. Invitees will be sent an email containing a link to the project.

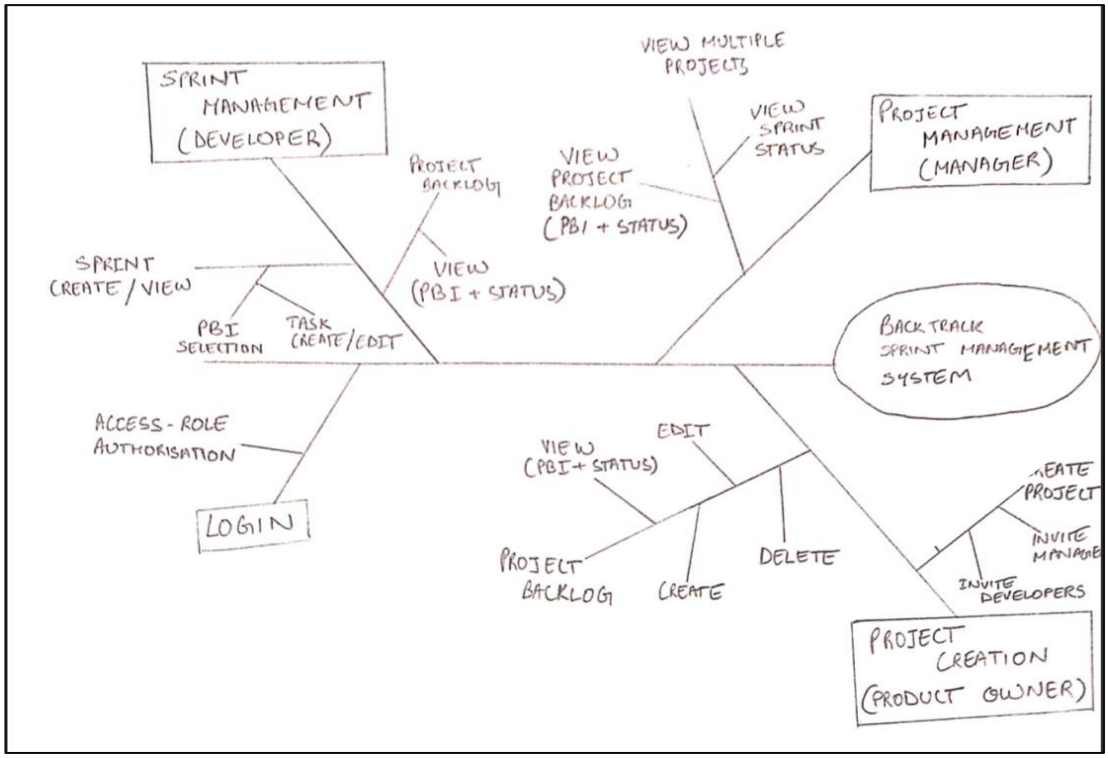

Fig. Product Feature Tree

2.2. Limitations and Exclusions

LI-1: As the content of a PBI may undergo change over time, Developer currently working on tasks under that PBI may not be aware of newer developments.
LI-2 The system doesn't take into account the case where two developers want to collaborate on a certain task under a PBI.
LI-3: Though each PBI in a Sprint has its own story-points and other attributes such as effort, nested individual items, once created by a developer, do not have story-point details or lengthy descriptions. LI-4. There is a lack of facility to handle the registration of a new user (Manager or Developer) to the database. This must be done manually.
LI-5. Further to ***LI-4,*** there is a shortcoming, that is, one cannot reset or modify account passwords, delete accounts or edit account details from the client side. It must be done manually as well.

3. Stakeholders

3.1. Stakeholder Profiles

Stakeholder	**Major Value**	**Major Interests**
Scrum Master / Manager	Improved productivity, streamlining business processes	Centralised system for managing teams, easy cross-team communication.

Product Owner	Improved productivity, easy ways of task allocation	Ease of forming teams for projects, centralised system for defining PBIs
Developer	Improved productivity	Better organisation of tasks. Reduced time required for communication Project status awareness
Client Company	Cost saving, automation of previously manual tasks, conformance to current standards or regulations	Reduced time in project development due to streamlining of teams, managers
HR	automation of previously manual tasks, improved productivity, ability to perform entirely new tasks (track activities of developers)	Easier to give performance reviews based on project activity

4. Delivery and Deployment

4.1. Product Roadmap

Feature	Release 1
FE - 1, Manager project-management page	Managers will be able to select the projects they manage and see the status of the product backlog.
FE - 2 Product Backlog Management	Product owner would be able to make modifications (create,edit,delete) to the project's product backlog.

FE - 3,4 Sprint Management	Developers can create and manage sprints. Add PBI from product backlog to the current sprint.Break down PBI into tasks and able to take ownership of those tasks.
FE - 5 View Product Backlog & Sprint	Everyone can view Product Backlog and Sprint (along with burndown chart and velocity).
FE - 6 System access	System access through authentication & authorization page (via internet)
FE - 7 Email Notices	Members to receive email upon invitation to project and given access link

4.2. Release Plan

Release 1:

Sprint 1 (8/10/19 - 28/10/19)	Sprint 2 (29/10/19 - 12/11/19)	Sprint 3 (13/11/19 - 26/11/19)
Product Backlog Management to be accessible by Product Owner **User authentication** will not be implemented yet **View Product Backlog (**Other team members**)**	**Sprint Creation** 1. Developers would be able to create sprints. 2. Developers will be able to move product backlog items to sprint. **Sprint Management** 1.Developers to be able to create tasks under each PBI. 2. Developers would be able to take ownership of tasks, change the status of completed tasks. 3. Developers can see the progress of the sprint.	**Manager Project Page** to be able to oversee multiple projects (Product backlog) **User Authentication** **Email Notice,Project Invites** **Deployment to the Heroku Server**

4.3. Deployment Considerations

A Web Server with the web application and a Database Management System will need to be setup, using a third party Web Service such as SendGrid for sending out emails. We will also be using private GitHub repositories for the frontend and backend of the application, separately. This would also help us to control the version of our application as well collaborating. The same repositories will be used by Heroku for deploying the frontend as well as the backend server for our web application. As this is only a prototype which will not require much bandwidth and computational power, the free version of Heroku will be sufficient. Depending on the bandwidth requirement in the future, we will switch to

Heroku standard or performance in the future. An up-to-date browser will be required along with a stable internet connection in order to access and use the web application.
Any possible infrastructure changes, if they occur, must be put in place at the time of the second release.