Software Requirements Specification

For

Agile-PDS

Version 0.1 draft

- Siddhant Misra

16-September-2016

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Siddhant Misra | 17-08-16 | Initial Draft | 0.1 |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

The purpose of this agile based PDS is to make easier the communication and interaction between employees, managers and customers.

This document will attempt to provide with all the basic needs for the development of a product with the help of agile.

**1.2 Document Conventions**

This document is based on IEEE standard with the font ‘ARIAL’ and heading size of 20, subheading size of 14 and content of size 12.

**1.3 Intended audience and reading suggestions**

This document is particularly intended for all the sectors like - IT, film industry, business management, software development companies, Board of directors, Managers, department hierarchies for maintaining the product’s development at different phases.

**1.4 Product scope**

Agile is expanding vastly in broader product development in sales and marketing departments. This change will help more and more industries of any genre to use this product and will facilitate better understanding of development to their customers and even their employees.

**1.5 References**

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements*

*Specifications.* IEEE Computer Society, 1998.

**2. Overall Description**

**2.1 Product Perspective.**

The Agile-PDS is a unique system which divides things in organizational hierarchy, and helps in forming the development timeline required for the completion of the product. The system is expected to evolve over several releases, ultimately connecting the customer and the firm to have a strong communication with each other leading towards a successful product, and to learn more about the product.

**2.2 Product Functions.**

Agile-PDS provides a user-interface to the firm’s manager and its customer to see the production timeline of the final product. It also provides an interface to the employees working under the product development team to input the further work progress of the modules. The customer can see the changes made to its product lively and can suggest any changes to be made through Agile-PDS chat-system/commenting, the respective changes would be made by the respective employee working under that module and it would be later reflected on the customer’s interface.The use case diagram in Figure 1 depicts the major groups of related requirements and how they work out in a particular scenario.

**2.3 User Cases and Characteristics.**

Customer An actor who defining the product specification to the

Manager/HRD, which in turn defines modules for the

product. Customer is also responsible for changes in modules and is

the only one in the system who is defining final objectives. Customer

has a given interface in which he first logs in through his credentials

and then would be able to see the product’s timeline, and if he wants to

implements some changes accordingly he can communicate with the

respective through the system’s chat/commenting service.

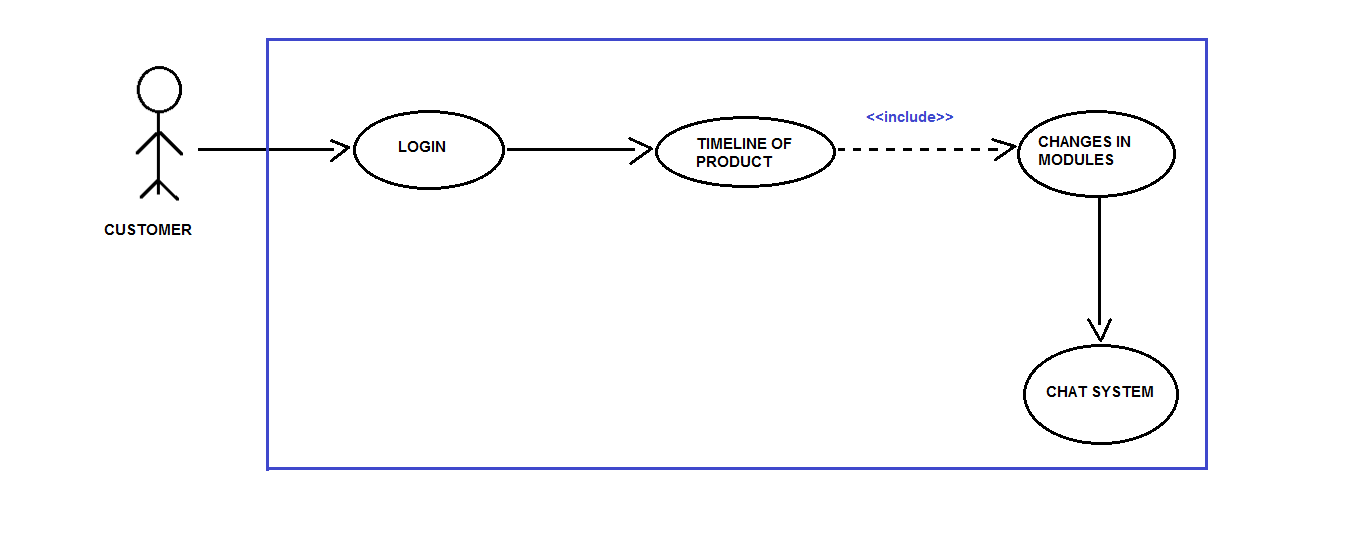


Figure 1.1

Manager/HRD

Responsible for the commencement of the product development phase, the manager/HRD first logs in with its credentials,

then can either view the earlier created tasks or creates a new

one. It then defines the modules (objectives) according to

the customer’s requirements of the product.

It is also responsible for the assignment of the modules to their respective

product development team(PDT).

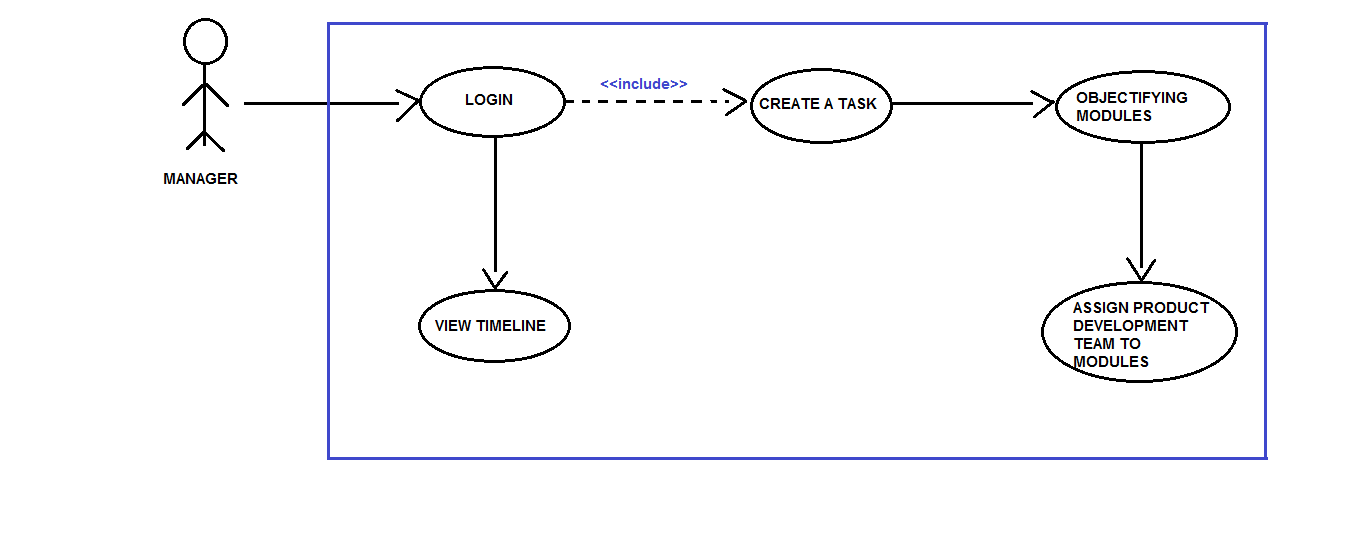


Figure 1.2

Employee The actor who is responsible for

the daily reporting on his assigned module by the

Manager/HRD. Employee is also responsible for the changes done to

the modules according to the customer. An employee can get its

objectives much clearer by communicating directly through the

Customer by Chat/Commenting service of the system.

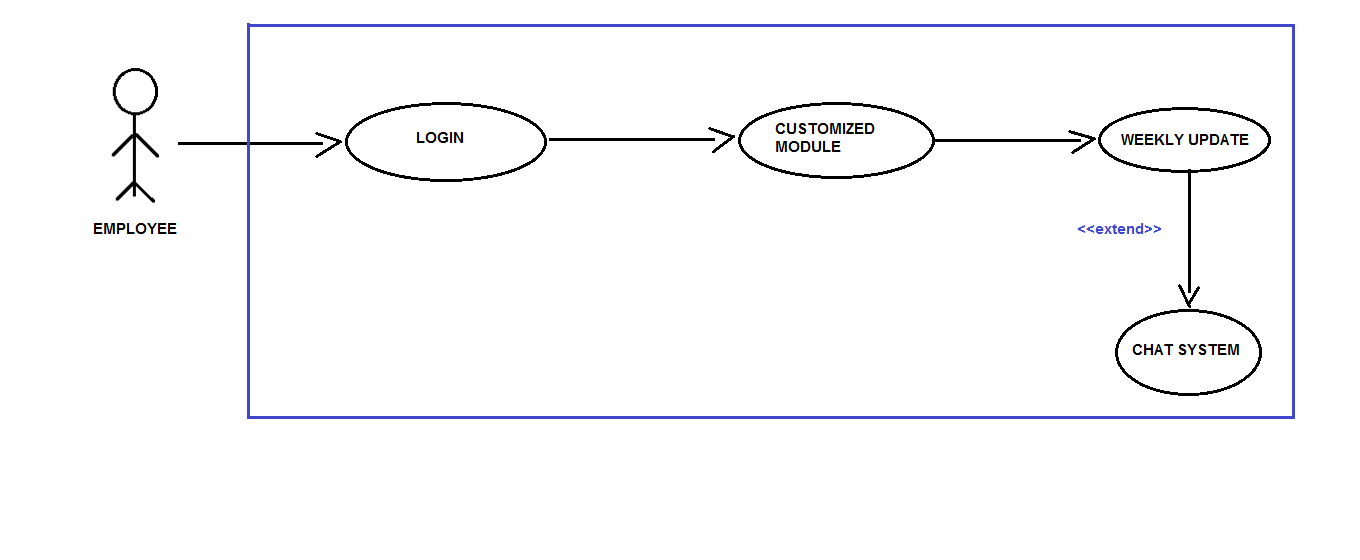


Figure 1.3

**2.4 Operating Environment.**

OE-1: The Agile-PDS shall operate with the following web browsers:

Microsoft Edge version-38 or latest, Google Chrome Version-53 or

Latest.

OE-2 The Agile-PDS shall operate on a server running the current corporate

approved versions of Jboss and Wildfly Web Server.

OE-3 The Agile-PDS shall permit employee access from the firm’s intranet,

and if the employee is allowed for the outside access through the firm’s

firewall,from an internet connection at employees remote location.

OE-4 The Agile-PDS shall have a hosted interface on the internet for the

customer.

**2.5 Design and Implementation Constraints.**

CON-1 The system shall use the current corporate standard of Mysql engine InnoDB.

CON-2 The system shall use the current corporate standard of Nosql with MongoDB.

CON-3 All HTML code shall conform to HTML 5.0 standard.

CON-4 All scripts shall be written in JavaScripts.

**2.6 User Documentation.**

UD-1 The system shall provide an online hierarchical and cross-linked help

system in HTML that describes and illustrates all system functions.

UD-2 The first time customer when accesses the system, on demand it will

be given Information regarding the User Interfaces.

**2.7 Assumptions and Dependencies.**

AS1 The final product made by the firm is feasible according to their

DE1 The operation of Agile-PDS depends on the employees working in

product development team inputting their work progress which helps in framing the timeline of the final product.

DE2 The operation of Agile-PDS depends on the customer in regular

commentation and communication about the final product’s

modules/goals.

**3. External Interface Requirements**

**3.1 User Interfaces:**

**UI 1:** The screen will display a login page with a dropbox, where there can be three different types of personnels entering their respective **username and password**.

**UI(for Customers):** The customers can see the progress and the timeline for the respective project. Also there will be a **chat system,** if there are any messages for any of the teams, the same can be conveyed through the chat system.

**UI(for manager):** The manager/HRD can create a new project, set the modules, assign teams to these modules and specify the timeline within which all this work is supposed to be done.

**UI(for Development team):** The personnel in the Development team can update their project progress/report, time to time. They can also send and revert messages through the **chat system**, if needed.

**3.2 Hardware Interfaces.**

Not identified yet.

**3.3 Software Interfaces.**

*SI1* Communicating Service.

*SI1.1*  Chat System is responsible for sending the comments provided by the

Customer to its designated employee working on that particular

module.

*SI1.2* Chat System can also be utilized by the employee in clearing its

objectives while designing/implementing its module.

**3.4 Communication Interfaces.**

*CI-1* The Agile-PDS should send an email to the customer on creation of the

task and about its modules.

*CI-2* The Agile-PDS should send an email to the customer to report any

problems with the modules(objectifying goals).

**4. System Features**

**4.1 Creating a Task**

**4.1.1 Description and Priority**

Agile PDS is one of the most stable platforms for development of any product or software. It helps ease and fasten the communication link between the manager and employees, employees and customers.

Priority = High.

**4.1.2 Functional Requirements**

|  |
| --- |
| **task.name:** The name of the task must be decided and shall be posted by manager whenever he wish to create his new task and shall be shared among the PDT.  **Task.customer\_name :** the name of the customer who has given the task must be written while creating the modules.  **Task.company\_name :** name of the company to which customer belongs shall be provided. |
| **Task.create\_modules :** manager shall create the modules based on the product specification which that company requires.  **Task.assignment \_of\_product\_to\_PDT :** Manager shall assign the task as per the specification of the team. |
| **Task.timeline :** Basically this will let customer to see the whole levels that till what stage the product is finished and which team is currently working at that moment.  **Task.changes\_in\_modules :**  Customer shall ask for some changes if its not according to his requirements.  **Task.chat\_system :** Using this chat system customer can give update about the changes whether it's okay or not and relating to his response the team can look after that query. |
| **Task.customized \_module :** employee shall work on the allotted module related to his profile.  **Task.weekly\_update :** employees should weekly update the modules on the basis of the **Task.chat\_system** according to the customer requirements. |

**4.2 Creating a module**

**4.3 Modify, view and delete module**

**4.4 Assigning modules to PDT**

**4.4.1 Description**

During this phase of development when all the modules are created and

designed then the team will be assigned to the modules who will work in the development process of that particular work. This team will also help in modifying the product on the weekly basis with respect to the changes

required by the customer after putting the request in the chat system box.

**5. Other Non-Functional Requirements**

To minimise the effort during development, we would need to involve the performance tester of our program, from the initial stages of planning. We would also be providing continuous feedback to developers, architects and system analysts, and at the same time sharing the performance test assets across the projects and versions.  
To ensure this runs smoothly we need to ensure that we have gathered all the performance related requirements and address them during discussions.

**5.1 Performance Requirements**

**Real Time Tracking:** The system would track the progress of all of the efforts that are being put into the project in real time. So, we would need to ensure that this system works properly at all stages, implying that it should continuously keep a track of the records of the developers and their contribution to this project.

**Transparency/Clarity of Communication:** This data would be made visible to the Human Resources Department, the Project Manager as well as the developers who are working on this project such that there is a hundred percent transparency rate during the workflow.  
**Grievance Redressal Mechanism:** The system would take the inputs from users and beta-testers as soon as the project is deployed, so if any user reports a bug or an issue, then it can be rectified then and there itself.   
**Reduction in the Bugs:** The primary objective being to keep the number of bugs and issues at bare minimum or debug the program as soon as an issue is brought to notice.

**5.2 Safety Requirements**

**Periodic Backups/Recovery** is required.

**5.3 Security Requirements**

**Preventing Data Leakage:** Data security and Privacy Protection is a high priority. Hence, a strong need would arise to protect the huge amount of data from being destroyed or manipulated by malicious users such as black-hat hackers and crackers.

**5.5 Business Rules**

* Everyone involved in this project, be it the business people or the developers will work together daily throughout the project, this in turn will not only help us achieve our goals earlier but more efficiently as well.
* Every individual must be self-reliant and self-directed, they should also have enough trust amongst fellow colleagues in order to complete the project.
* We would be open to implement changes at any given instance of time, whether the project is in its initial stages or the final stages.
* Customer satisfaction will be our highest priority, we will aim to achieve this through continuous delivery of the software at a regular interval of time.
* Regular meetings will be held to discuss the prospects of improvement at any stage of the project and choosing the right pace for the team members
* Lastly, we will keep things simple, minimized and organized at the same time. We will also keep a track of our progress at regular intervals