**AGILE DEVELOPMENT PLATFORM**

By

**ARCHIT DWIVEDI(1301010025)**

**ARCHANA PAL(1301010024)**

**CHETNA GUPTA(1301010041)**

Submitted to the Department of Computer Science & Engineering

In partial fulfilment of the requirements

For the degree of

Bachelor of Technology

In

Computer Science & Engineering



United College of Engineering & Research, Allahabad

A.P.J. Abdul Kalam Technical University, Lucknow

April, 2017

**TABLE OF CONTENTS**

DECLARATION

CERTIFICATE

ACKNOWLEDGEMENT

ABSTRACT

CHAPTER 1 INTRODUCTION

CHAPTER 2 SYSTEM ANALYSIS

2.1 Overview

2.2 Proposed System

2.3 Online Action

CHAPTER 3 SCRUM BOARDS AND TASK MANAGEMENT

CHAPTER 4 SYSTEM DESIGN

4.1 Terminology

4.2Data Flow Diagram

4.3 Use Case Diagram

4.4 Database Table Design

CHAPTER 5 TESTING

CHAPTER 6 OVERVIEW OF METEOR JS

CHAPTER 7 OVERVIEW OF REACT JS

CHAPTER 8 SCREESHOTS

CHAPTER 9 SYSTEM SPECIFICATION

8.1 Hardware Specification

8.2 Software Specification

CHAPTER 10 CONCLUSION

CHAPTER 11 REFERENCES

**DECLARATION**

We hereby declare that this submission is our own work and that to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to be substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning except where due acknowledgement has been made in the text.

**Signature:**

Name: Archit Dwivedi

Roll no: 1301010025

Date:

**Signature:**

Name: Archana Pal

Roll no: 1301010024

Date:

**Signature:**

Name: Chetna Gupta

Roll no: 1301010041

Date:

**CERTIFICATE**

It is certified that **Archit Dwivedi, Archana Pal, Chetna Gupta** has carried out the research work presented in this project entitled “**AGILE DEVELOPMENT PLATFORM**”, for the award of Bachelor of Technology from A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The project embodies result of original work and studies carried out by students themselves and the contents of the project do not form the basis for the award of any other degree to the candidate or to anybody else.

**Date:**

**Supervisor:**

**Mr. O.P AGRAHARI**

Assistant Professor

(Dept. of C.S.E)

**ACKNOWLEDGEMENT**

It is with great sense of gratitude that we acknowledge the support given to us by number of individuals in the preparation of development of this project. There are lots of people who shaped our thinking and encouraged us to do this project.

We owe special dept of gratitude to Mr.Sanjay Pandey, Department of Computer Science & Engineering at United College of Engineering & Research, for his constant support and guidance throughout the course of our work. His sincerely, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant effort that our endeavours have seen light of the day.

We would like to express our deep gratitude to Mr. Prabhat Shukla for being our guide and helping us throughout the project with the words of encouragement and showing full confidence in our ability. We admire his “can-do” attitude.

At last but not the least we cannot forget to give special thanks to our friends and batch-mates and all faculty members of Computer Science department.

**Signature:**

Name: Archit Dwivedi

Roll no: 1301010025

Date:

**Signature:**

Name: Archana Pal

Roll no: 1301010024

Date:

**Signature:**

Name: Chetna Gupta

Roll no: 1301010041

**Date:**

**ABSTRACT**

Agile is a software development methodology to build a software incrementally using short iterations of 1 to 4 weeks so that the development process is aligned with the changing business needs. Instead of a single-pass development of 6 to 18 months where all the requirements and risks are predicted upfront, Agile adopts a process of frequent feedback where a workable product is delivered after 1 to 4 week iteration.

We will be using **SCRUM** for Methodology for developing our platform.

The definition of **done** for User Story, Iteration, and Release is given below.

## User Story

A user story is a requirement which is formulated in a few sentences in everyday language of an user and it should be completed within an iteration. A user story is done when

* All the related code have been checked-in.
* All the unit test cases have been passed.
* All the acceptance test cases have been passed.
* Help text is written.
* Product Owner has accepted the story.

## Iteration

An iteration is a time boxed collection of user stories / defects to be worked upon and accepted within the release of a product. Iterations are defined during iteration planning meeting and completed with an iteration demo and review meeting. An iteration is also termed as a **sprint**.

**CHAPTER 1**

**INTRODUCTION**

**What is agile?**

Agile is a software development methodology to build a software incrementally using short iterations of 1 to 4 weeks so that the development process is aligned with the changing business needs. Instead of a single-pass development of 6 to 18 months where all the requirements and risks are predicted upfront, Agile adopts a process of frequent feedback where a workable product is delivered after 1 to 4 week iteration.  
  
We are developing our platform on the basis of scrum methodology.  
  
**What is Scrum?**

Scrum is an efficient framework within which you can develop software with teamwork. It is based on agile principles.

**CHAPTER 2**

**SYSTEM ANALYSIS**

# 2.1 OVERVIEW

## Scrum Definition

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Scrum is a process framework that has been used to manage complex product development since the early 1990s. Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and development practices so that you can improve.

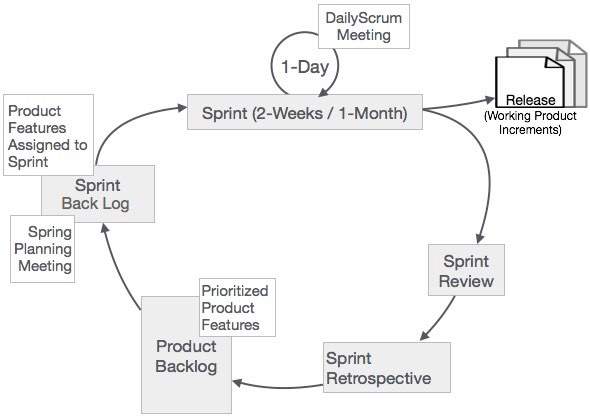
The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum’s success and usage.

The rules of Scrum bind together the events, roles, and artifacts, governing the relationships and interaction between them. The rules of Scrum are described throughout this tutorial.

**Note** - Across the industry, there are misconceptions that Scrum means no documentation, scrum team consists of only developers, and so on. It is not entirely so; we will give clarifications on these in later sections.

# 2.2 PROPOSED SYSTEM

## Scrum Process Framework



In Scrum, the prescribed events are used to create regularity. All events are time-boxed events, such that every event has a maximum duration. The events are described more elaborately in the subsequent chapters.

## Sprint

The heart of Scrum is a Sprint, a time-box of two weeks or one month during which a potentially releasable product increment is created. A new Sprint starts immediately after the conclusion of the previous Sprint. Sprints consist of the Sprint planning, daily scrums, the development work, the Sprint review, and the Sprint retrospective.

* In Sprint planning, the work to be performed in the Sprint is planned collaboratively by the Scrum Team.
* The Daily Scrum Meeting is a 15-minute time-boxed event for the Scrum Team to synchronize the activities and create a plan for that day.
* A Sprint Review is held at the end of the Sprint to inspect the Increment and make changes to the Product Backlog, if needed.
* The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. In this meeting, the Scrum Team is to inspect itself and create a plan for improvements to be enacted during the subsequent Sprint.

Scrum is a process framework that defines certain rules, events, and roles to bring in regularity. However, it can be adapted to any organization, based on needs, provided the basic scrum rules are not violated.

**CHAPTER 3**

**SCRUM BOARDS AND TASK MANAGEMENT**

**ROLES:**

**The Scrum Team consists of three roles, namely a ScrumMaster, a Product Owner, and the Team.**

## ScrumMaster

The ScrumMaster (sometimes written as the Scrum Master, although the official term has no space after “Scrum”) is the keeper of the scrum process. He/she is responsible for-

* making the process run smoothly
* removing obstacles that impact productivity
* organizing and facilitating the critical meetings

## Product Owner

The Product Owner is responsible for maximizing the value of the product and the work of the Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes-

* Expressing Product Backlog items clearly.
* Ordering the Product Backlog items to best achieve goals and missions.
* Optimizing the value of the work the Team performs.
* Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Team will work on further.
* Ensuring that the Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Team do it. However, the Product Owner remains accountable for these tasks.

The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item’s priority must address the Product Owner.

For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner’s decisions are visible in the content and ordering of the Product Backlog. No one is allowed to tell the Team to work from a different set of requirements, and the Team is not allowed to act on what anyone else says. This is ensured by ScrumMaster.

## The Team

The Team is self-organizing and cross-functional. That means the team comprises of analysts, designers, developers, testers, etc. as appropriate and as relevant to the project.

Some people in the industry refer to this team as development team. However, such a reference is leading to controversy that the team can have only developers and no other roles. It is an obvious understanding that it is only a misconception. To develop a software product, we require all the roles and that is the essence of scrum – the team will function in collaboration. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team, and thus time and effort can be saved. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.

Optimal Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. The Team size should be kept in the range from five to nine people, if possible. Fewer than five team members decrease interaction and results in smaller productivity gains. Having more than nine members requires too much coordination.

The scrum team works together closely, on a daily basis, to ensure the smooth flow of information and the quick resolution of issues. The scrum team delivers product iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of a complete product ensure a potentially useful version of working product is always available.

**CHAPTER 4**

# SYSTEM DESIGN

Scrum Process Framework can be viewed by means of a sequence of events and the corresponding artifacts. The Scrum events are time-boxed events. That means, in a project, every scrum event has a predefined maximum duration. These events enable transparency on the project progress to all who are involved in the project. The vital events of scrum are-

* The Sprint
* Sprint Planning
* Daily Scrum Meetings
* The Sprint Review
* The Sprint Retrospective

## The Sprint

During a Sprint, a working product Increment is developed. It is usually of duration two weeks or one month, and this duration remains constant for all the sprints in the project. We cannot have varying durations for the different sprints in a project. A new Sprint starts immediately after the conclusion of the previous Sprint.

The Sprint Goal is an objective set for the Sprint. It provides guidance to the Team on why it is building the Increment. It is created during the Sprint Planning meeting. The scope of the sprint is clarified and re-negotiated between the Product Owner and the Team as more about the requirements is learned. Thus, each Sprint is associated with it, a definition of what is to be built, a design, and the flexible plan that will guide building it, the development work, and the resultant product increment.

A Sprint should be cancelled if the Sprint Goal becomes obsolete. This might occur if the organization changes direction or if market or technology conditions change. A sprint can be cancelled only by product owner, though others have an influence on the same.

Due to the short duration nature of Sprints, cancellation during a sprint rarely makes sense. As the sprint cancellations consume resources, for getting re-organized into another Sprint, they are very uncommon.

If a Sprint is cancelled, and part of the work produced during the sprint is potentially releasable, the Product Owner typically accepts it. All the incomplete Sprint Backlog Items are put back into the Product Backlog.

## Sprint Planning

The work to be performed in the Sprint is planned in the Sprint Planning Meeting. Sprint Planning Meeting is of duration of maximum of four hours for two weeks sprints and eight hours for one month Sprints. It is the responsibility of the Scrum Master to ensure that the meeting takes place and that all the required attendees are present and understand the purpose of the scheduled meeting. The Scrum Master moderates the meeting to monitor the sustenance of discussion and closure on time.

Sprint Planning focuses on the following two questions -

* What needs to be and can be delivered in the Sprint Increment?
* How will the work needed for the execution of Sprint be achieved?

The inputs to this meeting are -

* The Product Backlog
* The latest product Increment
* Projected capacity of the Team during the Sprint
* Past performance of the Team

The Scrum Team discusses the functionality that can be developed during the Sprint. Product Owner provides clarifications on the Product Backlog items. The team selects the items from the Product Backlog for the Sprint, as they are the best to assess what they can accomplish in the Sprint. The Team comprises of analysts, designers, developers, and testers. The work is carried out in a collaborative fashion, thus minimizing re-work.

The Scrum Team then comes up with Sprint Goal. The Sprint Goal is an objective that provides guidance to the Team on why it is building the Product Increment. The Team then decides how it will build the selected functionality into a working product Increment during the Sprint. The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog.

Work during a sprint is estimated during sprint planning and may be of varying size and/or effort. By the end of the Sprint Planning meeting, the work is divided into tasks of duration of one day or less. This is to enable the ease of work allocation, and tracking the completion. If the Team realizes that it has too much or too little work, it can renegotiate the selected Product Backlog items with the Product Owner.

The Team may also invite others (not part of Scrum Team) to attend the Sprint Planning meeting to obtain technical or domain advice or help in estimation.

## Daily Scrum Meetings

The Daily Scrum Meeting is a 15-minute meeting for the Team, conducted daily to quickly understand the work since the last Daily Scrum Meeting and create a plan for the next 24 hours. This meeting is also referred to as Daily Stand up Meeting.

The Daily Scrum Meeting is held at the same time and same place every day to reduce complexity.

During the meeting, each Team member explains -

* What did he do yesterday that helped the Team meet the Sprint Goal?
* What will he do today to help the Team meet the Sprint Goal?
* Does he see any impediments that prevent him or the Team from meeting the Sprint Goal?

Daily Scrum is mistaken to be a status tracking event, though, in fact, it is a planning event.

The input to the meeting should be how the team is doing toward meeting the Sprint Goal, and the output should be a new or revised plan that optimizes the team’s efforts in meeting the Sprint Goal.

Though the Scrum Master coordinates the Daily Scrum Meeting and ensures that the objectives of the meeting are met, the Meeting is the responsibility of the Team.

If necessary, the Team may meet immediately after the Daily Scrum Meeting, for any detailed discussions, or to re-plan the rest of the Sprint’s work.

Following are the benefits of Daily Scrum Meetings -

* Improve communication within the Team.
* Identify impediments, if any, in order to facilitate an early removal of the same, so as to minimize impact on the Sprint.
* Highlight and promote quick decision-making.
* Improve the Team’s level of knowledge.

## Sprint Review

A Sprint Review is held at the end of every Sprint. During the Sprint Review, a presentation of the increment that is getting released is reviewed. In this meeting, the Scrum Team and the stakeholders collaborate to understand what was done in the Sprint. Based on that, and any changes to the Product Backlog during the Sprint, the attendees arrive at the next steps required that could optimize value. Thus, the objective of Sprint Review is to obtain feedback and progress unitedly.

The Sprint Review is normally held for two hours for two week sprints and for four hours for one month sprints.

The Scrum Master ensures that -

* The meeting takes place.
* The participants understand the purpose.
* The meeting is focused on the required agenda and is completed within the required duration.

The Sprint Review includes the following aspects -

* Attendees include the Scrum Team and key stakeholders, as invited by the Product Owner.
* The Product Owner explains what Product Backlog items have been completed during the sprint and what has not been completed.
* The Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved.
* The Team demonstrates the work that it has completed and answers questions, if any, about the Increment.
* The entire group then discusses on what to do next. Thus, the Sprint Review provides valuable input to Sprint Planning of the subsequent Sprint.
* The Scrum Team then reviews the timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product increment.
* The outcome of the Sprint Review is an updated Product Backlog, which defines the probable Product Backlog items for the next Sprint.

## Sprint Retrospective

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is usually a one hour meeting for two-week duration sprints and a three hour meeting for one month duration Sprints.

The purpose of the Sprint Retrospective is to -

* Combine the learnings from the last Sprint, with regards to people, relationships, process, and tools.
* Identify the major items that went well and potential improvements.
* Creation of a plan for implementing improvements to increase product quality.

The Sprint Retrospective is an opportunity for the Scrum Team to introspect and improve within the Scrum process framework so as to make the next Sprint outcome more effective.

# 4.1 TERMINOLOGY

Cryptography: Cryptography is the study of encryption. This is a process for coding data so unauthorized people cannot understand it. Many products use encryption to code information. There are two different methods for encrypting data. One way is asymmetric. This requires a public key to encrypt the data and a private key to decrypt. The alternative is through symmetric encryption. This uses a private key to encrypt and decrypt the information. Encryption is done through the use of algorithms.

Firewalls: Firewalls can be implemented through either hardware or software. It filters the incoming and outgoing packets and determines if it wants to transport them to their destination. It prohibits unauthorized connections into the network from certain ports.

Smart Cards: Smart Cards are small credit card like products that contain a small microchip on them. These microchips store data. The cards can be either disposable or reloadable. Reloadable cards have the option to add additional information or applications at a later time. Cards are usually loaded with cybercash, encryption information, and much more.

Digital Signatures: Digital Signatures are used to ensure that only authorized people are able to read a certain message. It is used as an electronic substitute to a manual signature. It consists of a sequence of bits that are encrypted with a sender's private key. When the recipient receives the message it is decrypted by the use of their public key. This enables the receiver to verify who sent it and whether the data has been altered in any way. The use of a time stamp can be used with the digital signature to confirm the time of the signature.

Digital Certificates: Digital Certificates have encoded personal information about the owner of the certificate. It is similar to Digital signatures in it allows the recipient of the message to verify the authenticity of the message. It secures information between browsers and servers, customer and merchants, or between two email partners. A third party, known as a certificate authority, distributes them. They are tamper proof and cannot be forged.

Common standards used in business transactions:

EDI- Electronic Data Interchange: It allows for transmission of information between private networks.

OBI- Open Buying on the Internet: Formed by the Internet Purchasing Roundtable, OBI makes communication possible between all ecommerce systems.

OTP- Open Trading Protocol: It's the standardization of payment transactions and was created to compete against the OBI standard.

OPS- Open Profiling Standard: It holds information about users likes and interests and distributes it to merchants.

SSL- Secure Socket Layer: This allows data to be transmitted securely by use of public key encryption. It was created by Netscape and is now public domain.

SET- Secure Electronic Transactions: Created by Visa and MasterCard, it encrypts credit card numbers located on servers.

**4.2 DATA FLOW DIAGRAMS**

Data flow diagram is a structure analysis tool that is used for graphical representation of Data processes through any organization . the data flow approach emphasis on the logic underlying the system, by using combination of only 4 symbols. It follows a top down approach. A full description of a system actually consists of set of DFD s , which comprises of various levels. And initial over view model is exploded lower level diagrams that show additional feature of the system. Further each process can be broken down into a more detailed DFD. This occurs repeatedly until sufficient details is described.

**DFD symbols**

**Square**

it defines a source (originator ) or destination of system data.

**Arrow**

It indicates data flow-data in motion. It is a pipeline through which information flows.

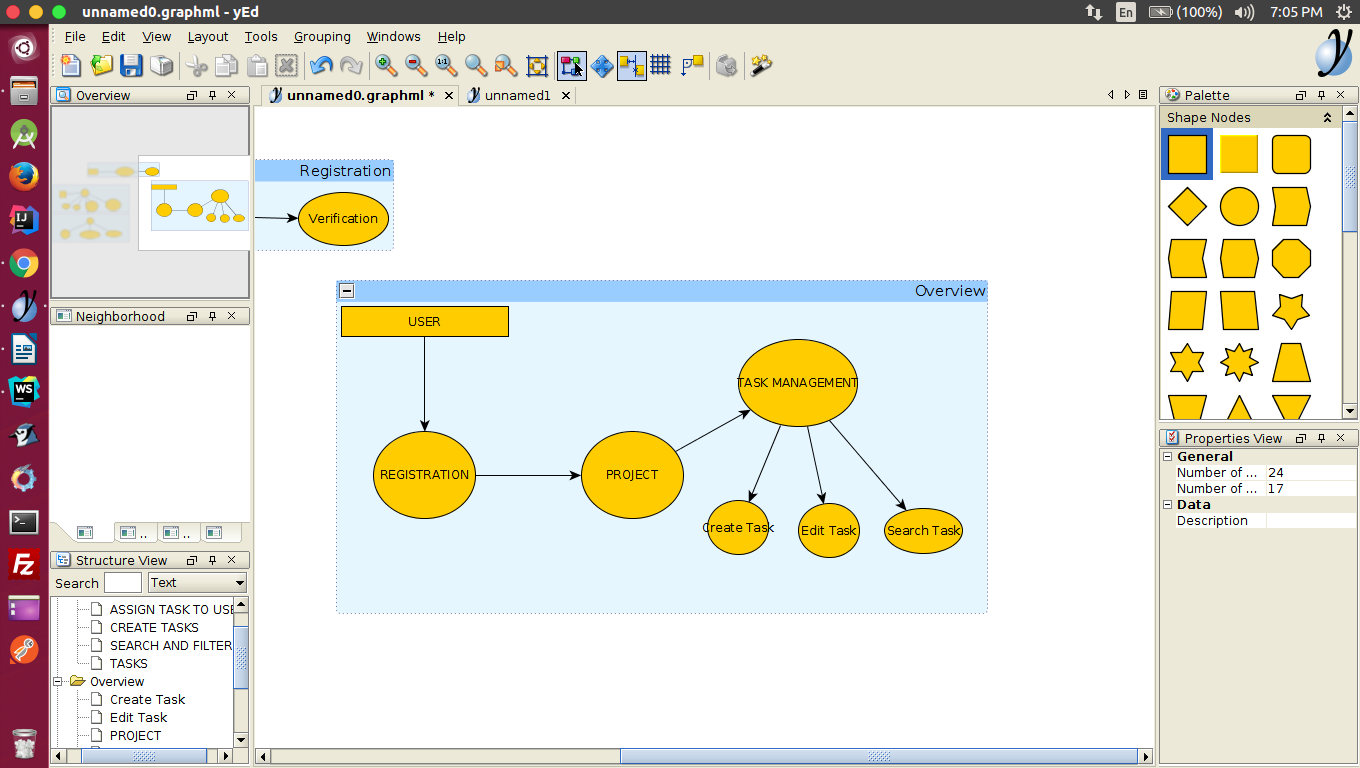
**Circle or Bubble**

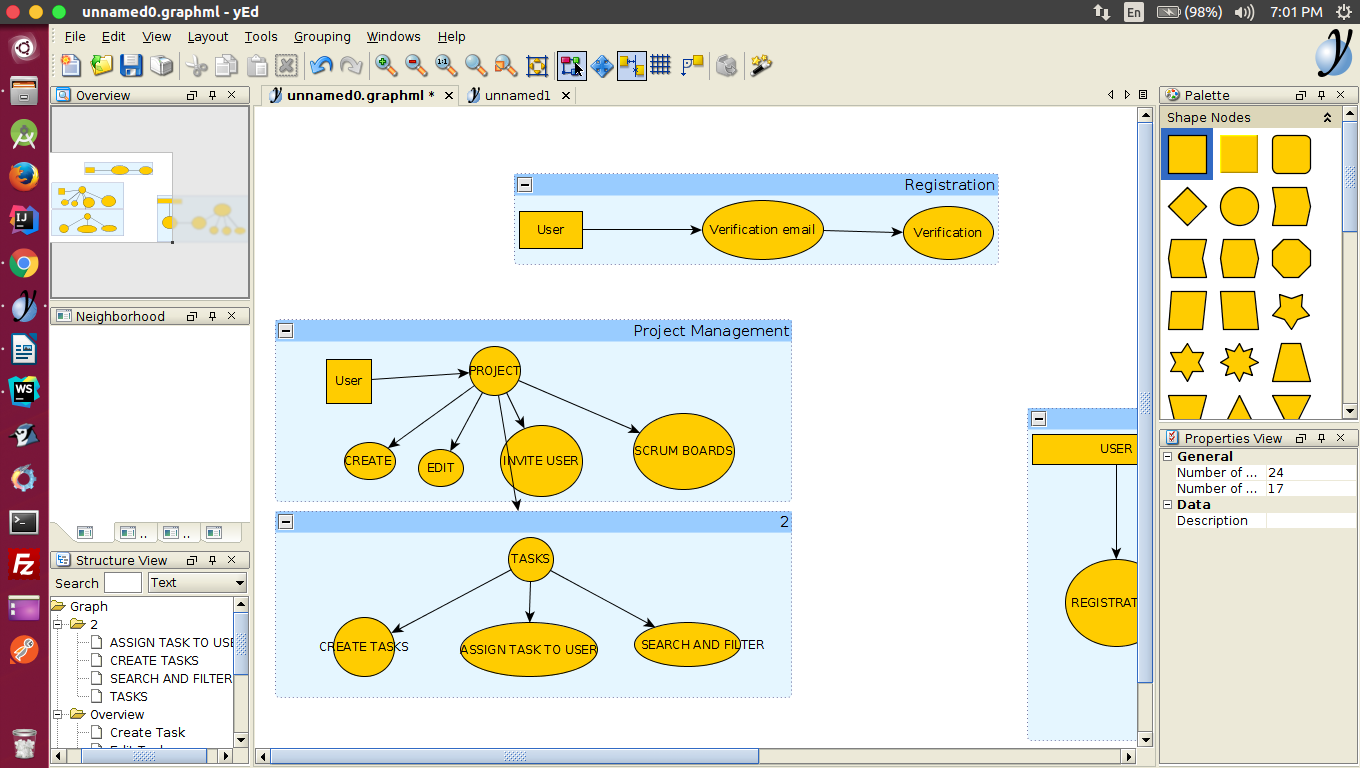
It represents a process that transforms incoming data flow(s) into outgoing data flow(s).

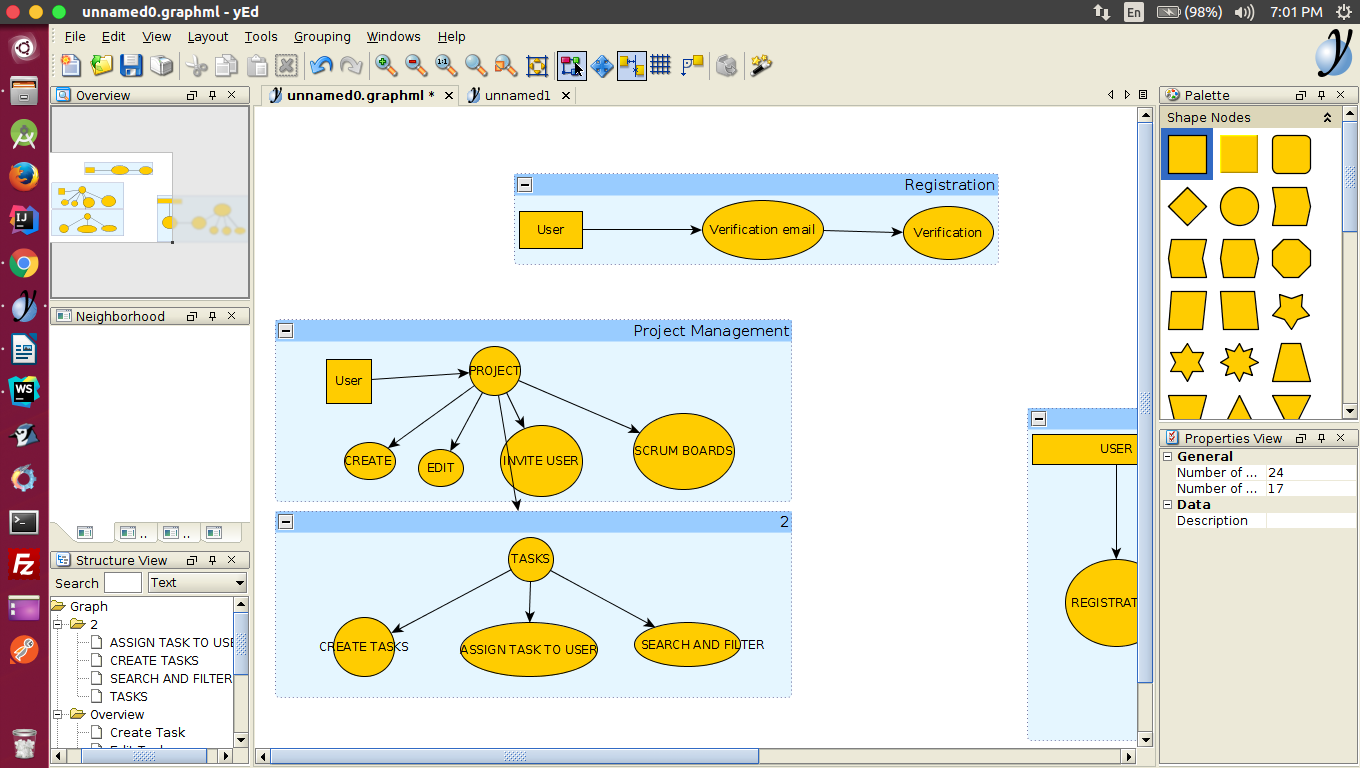
**Open Rectangle**

****

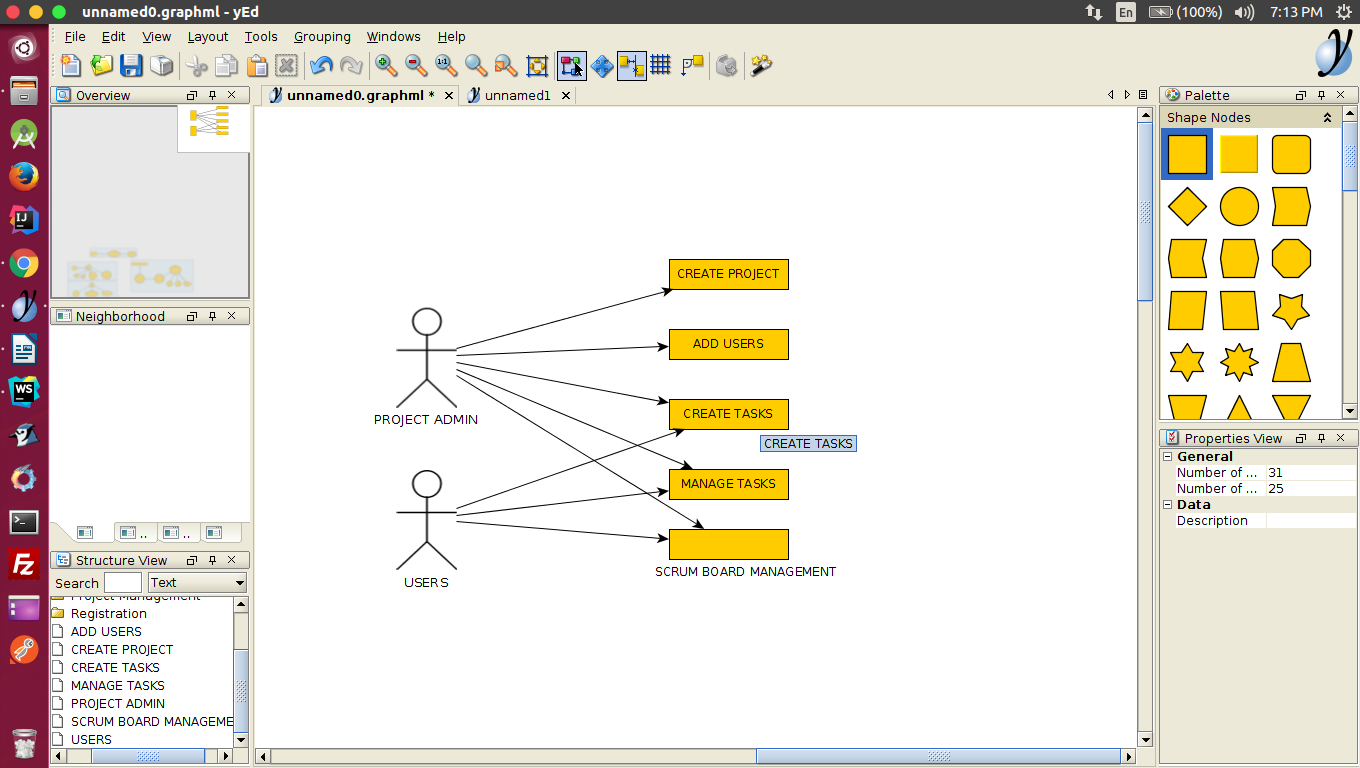
it is a data store-data at rest, or a temporary repository of data. Here I am giving only the Data Flow Diagram.

****



****

**4.3 USE CASE DIAGRAM**



**4.4 DATABASE TABLE DESIGN**

DATABASE USED: MONGO-DB

**DBPROJECTS : PROJECTS COLLECTION**

{

"\_id" : "FoWDhboAKQuuyCEdi",

"projectName" : "Final Year Project",

"description" : "This project is for final year project",

"createdOn" : "Tue Apr 25 2017 19:08:31 GMT+0530 (IST)",

"users" : [

"tcmt7HfryowX5Qv5K"

],

"createdBy" : "tcmt7HfryowX5Qv5K",

"scrumBoards" : [

"Design",

"Coding",

"Testing"

],

"notifications" : [

{

"message" : "Created Project",

"updatedOn" : "Tue Apr 25 2017 19:08:31 GMT+0530 (IST)",

"updatedBy" : "tcmt7HfryowX5Qv5K"

},

{

"message" : "Scrum Board Added: Design",

"updatedOn" : "Tue Apr 25 2017 19:09:02 GMT+0530 (IST)",

"updatedBy" : "tcmt7HfryowX5Qv5K"

},

{

"message" : "Scrum Board Added: Coding",

"updatedOn" : "Tue Apr 25 2017 19:09:05 GMT+0530 (IST)",

"updatedBy" : "tcmt7HfryowX5Qv5K"

},

{

"message" : "Scrum Board Added: Testing",

"updatedOn" : "Tue Apr 25 2017 19:09:07 GMT+0530 (IST)",

"updatedBy" : "tcmt7HfryowX5Qv5K"

}

]

}

**USERS COLLECTION**

{

"\_id" : "YCxN3QjLrvceCGeyg",

"createdAt" : ISODate("2017-04-23T07:58:31.798Z"),

"services" : {

"password" : {

"bcrypt" : "$2a$10$f4uXDpOWZhxoCSZhnb9wI.BJZhNxuGHUi9fFSy90u7ac5q0PFFsk6"

},

"resume" : {

"loginTokens" : [

{

"when" : ISODate("2017-04-23T07:58:31.804Z"),

"hashedToken" : "ezCDW/VfLcno7gWiUktrTE+cTDOcbIcp66VskG6CR8k="

}

]

}

},

"emails" : [

{

"address" : "asd@gmail.com",

"verified" : true

}

],

"name" : "Archit"

}

**DB TASKS COLLECTIONS**

{

"\_id" : "2",

"title" : "kjlkjb",

"description" : "kjb",

"assignee" : "2dwzzEa995hGZAmNa",

"status" : "In-Progress",

"scrumBoard" : "UI Design",

"percentDone" : "0",

"parentTask" : "",

"projectId" : "nbQmTWCovqchPXvCz",

"createdBy" : "2dwzzEa995hGZAmNa",

"createdOn" : "Sat Apr 22 2017 11:18:59 GMT+0530 (IST)",

"notifications" : [

{

"message" : "Created Task",

"updatedOn" : "Sat Apr 22 2017 11:18:59 GMT+0530 (IST)",

"updatedBy" : "2dwzzEa995hGZAmNa"

}

]

}

**CHAPTER 5**

**TESTING**

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

1. **Unit Testing**

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

**Each module can be tested using the following two strategies:**

1. **Black Box Testing:**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structure or external database access
4. Performance errors
5. Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

1. **White Box testing**

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

It has been uses to generate the test cases in the following cases:

Guarantee that all independent paths have been executed.

Execute all logical decisions on their true and false sides.

Execute all loops at their boundaries and within their operational bounds.

Execute internal data structures to ensure their validity.

1. **Integrating Testing**

Integration testing ensures that software and subsystems work together as a whole. It tests the interface ofall the modules to make sure that the modules behave properly when integrated together.

**3. System Testing**

Involves in-house testing of the entire system before delivery to the user. It's aim is to satisfy the user the system meets all requirements of the client's specifications.

1. **Acceptance Testing**

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

**Validation**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

**CHAPTER 6**

**OVERVIEW OF METEOR JS**

**According to Meteor official documentation −**

Meteor is a full-stack JavaScript platform for developing modern web and mobile applications. Meteor includes a key set of technologies for building connected-client reactive applications, a build tool, and a curated set of packages from the Node.js and general JavaScript community.

## Meteor Features

* **Web and Mobile** − Meteor offers platform for developing Web, Android and IOS apps.
* **Universal Apps** − The same code for web browsers and mobile devices.
* **Packages** − Huge number of packages easy to install and use.
* **Meteor Galaxy** − Cloud service for Meteor app deployment.

## Meteor Advantages

* Developers only need JavaScript for server and client side development.
* The coding is very simple and beginner friendly.
* Meteor apps are real time by default.
* Official and community packages are huge time saver.

## Meteor Limitations

* Meteor isn't very suitable for large and complex application.
* There is a lot of magic going on when working with Meteor so developers might find themself limited in some way.

**CHAPTER 7**

**OVERVIEW OF REACT JS**

Definition can be found on React official documentation −

React is a library for building composable user interfaces. It encourages the creation of reusable UI components which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.

## React Features

* **JSX** − JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended.
* **Components** − React is all about components. You need to think of everything as a component. This will help you to maintain the code when working on larger scale projects.
* **Unidirectional data flow and Flux** − React implements one way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.
* **License** − React is licensed under the Facebook Inc. Documentation is licensed under CC BY 4.0.

## React Advantages

* React uses virtual DOM which is JavaScript object. This will improve apps performance since JavaScript virtual DOM is faster than the regular DOM.
* React can be used on client and server side.
* Component and Data patterns improve readability which helps to maintain larger apps.
* React can be used with other frameworks.

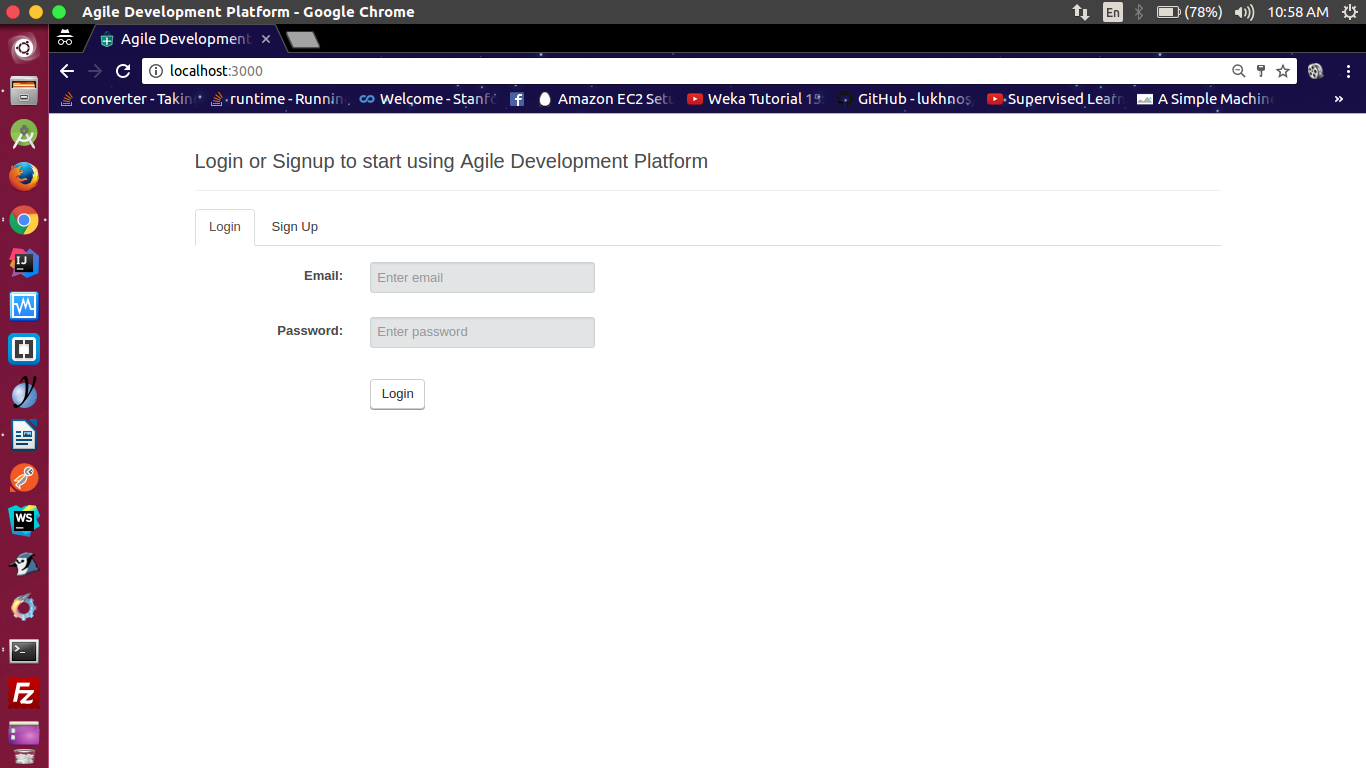
## React Limitations

* React only covers view layer of the app so you still need to choose other technologies to get a complete tooling set for development.
* React is using inline templating and JSX. This can seem awkward to some developers.

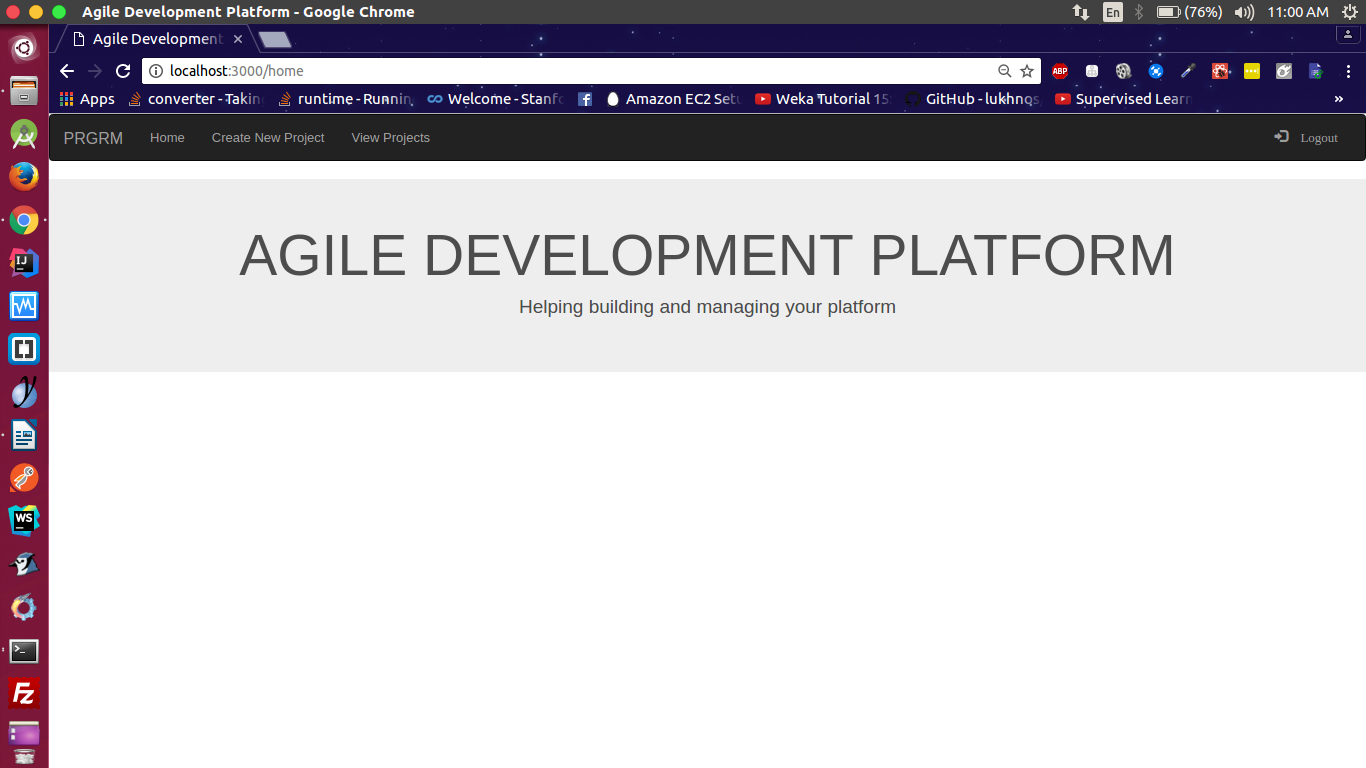
**CHAPTER 10**

**SCREENSHOTS**

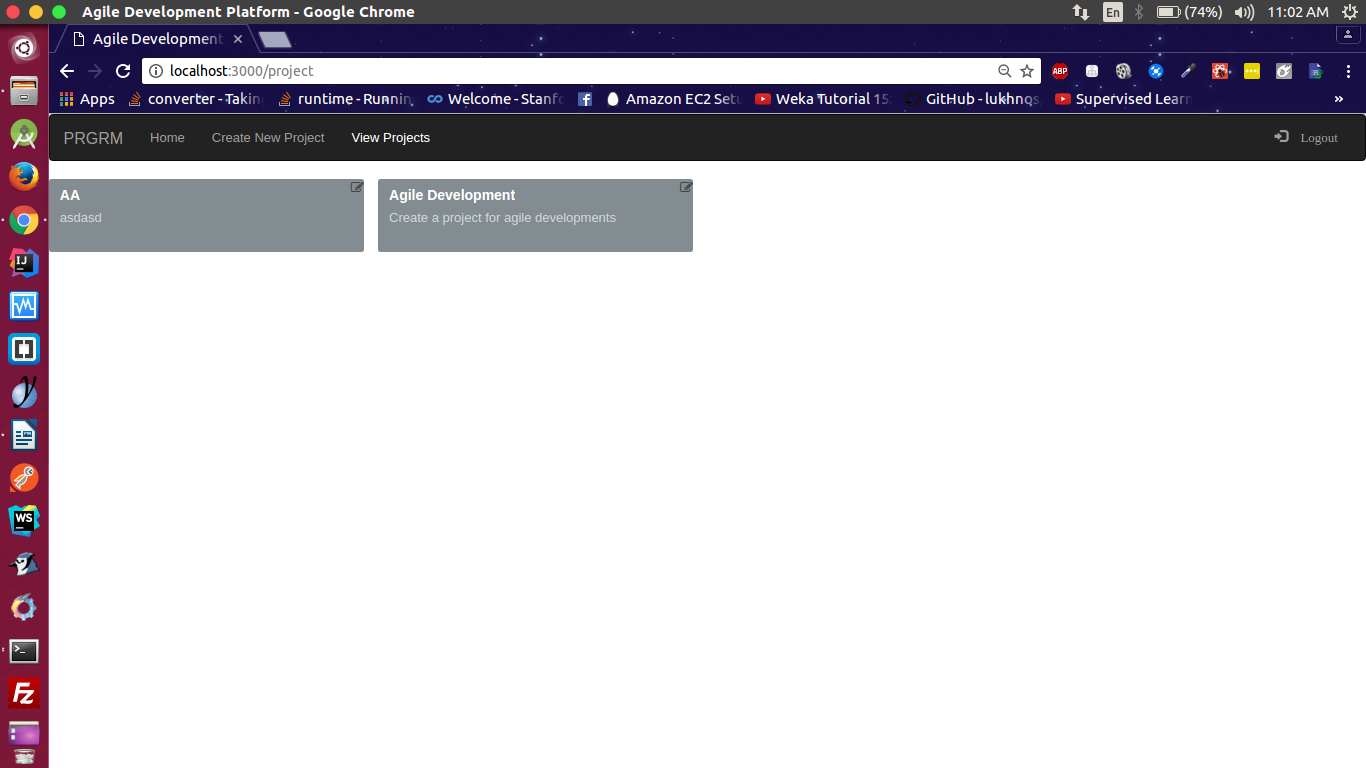
**1: Login Signup**



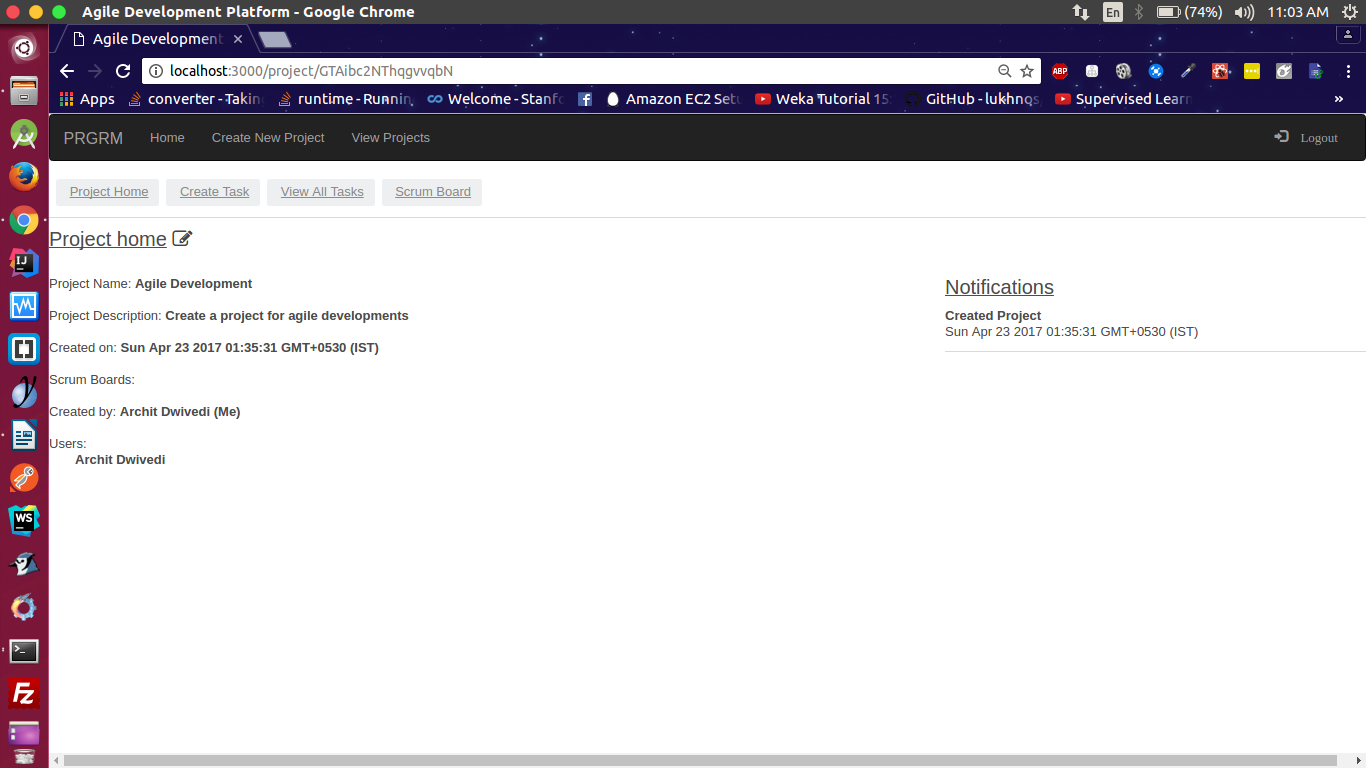
**2: Home Page**



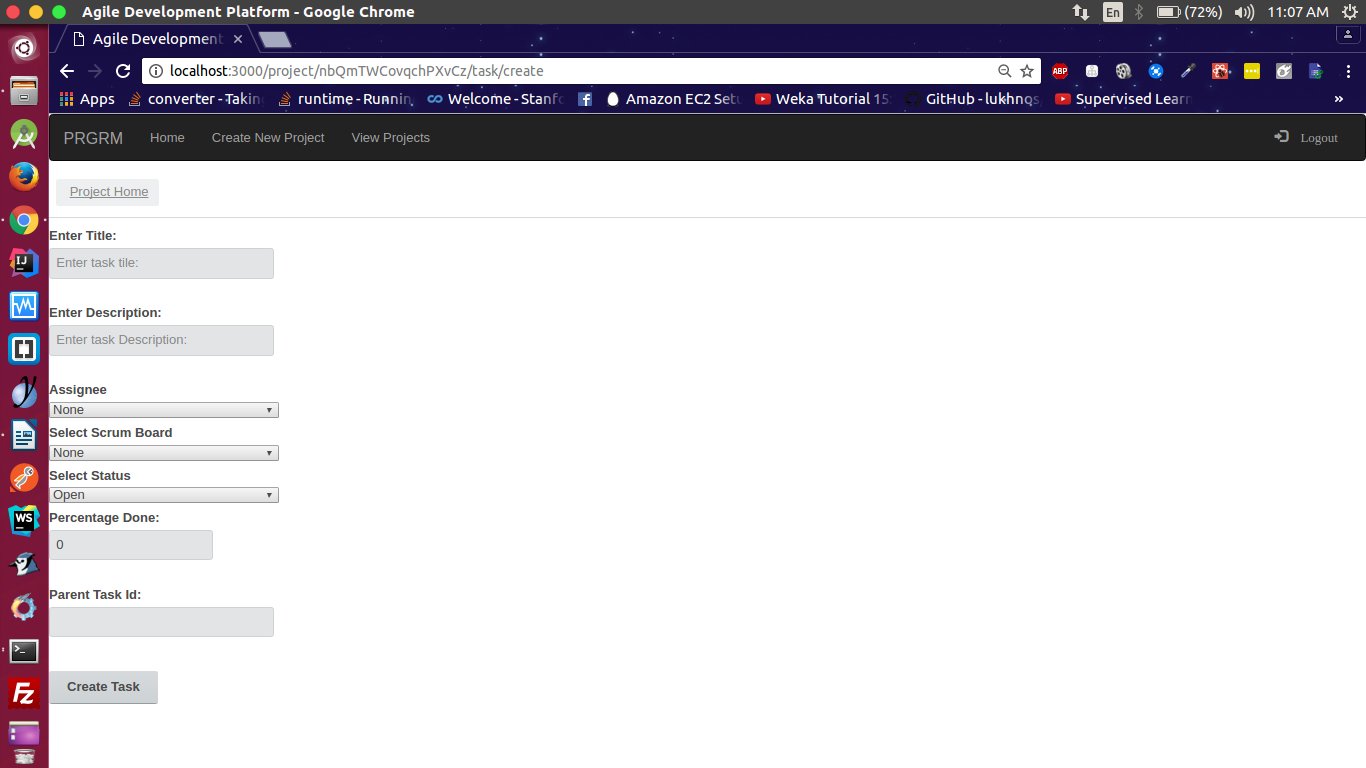
**3: Project Page**



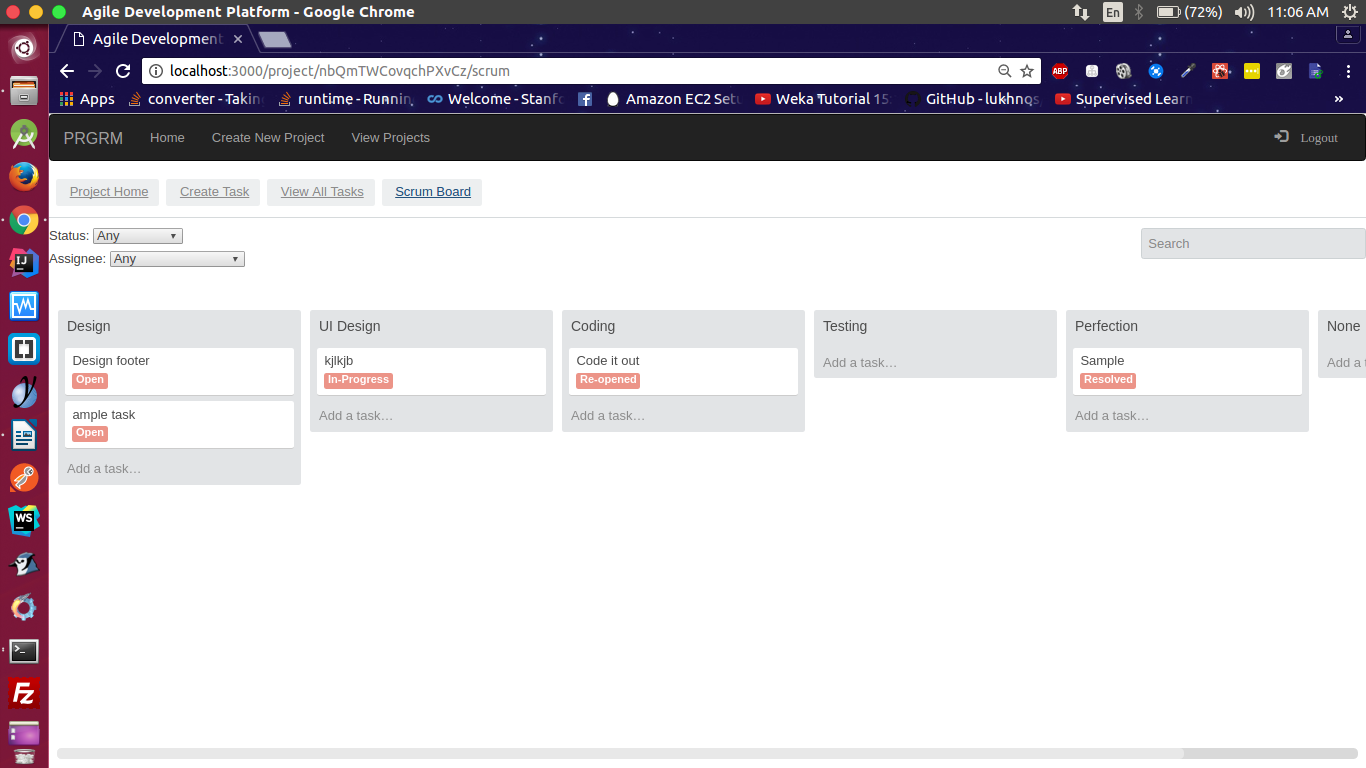
**4: Project Home Page:**



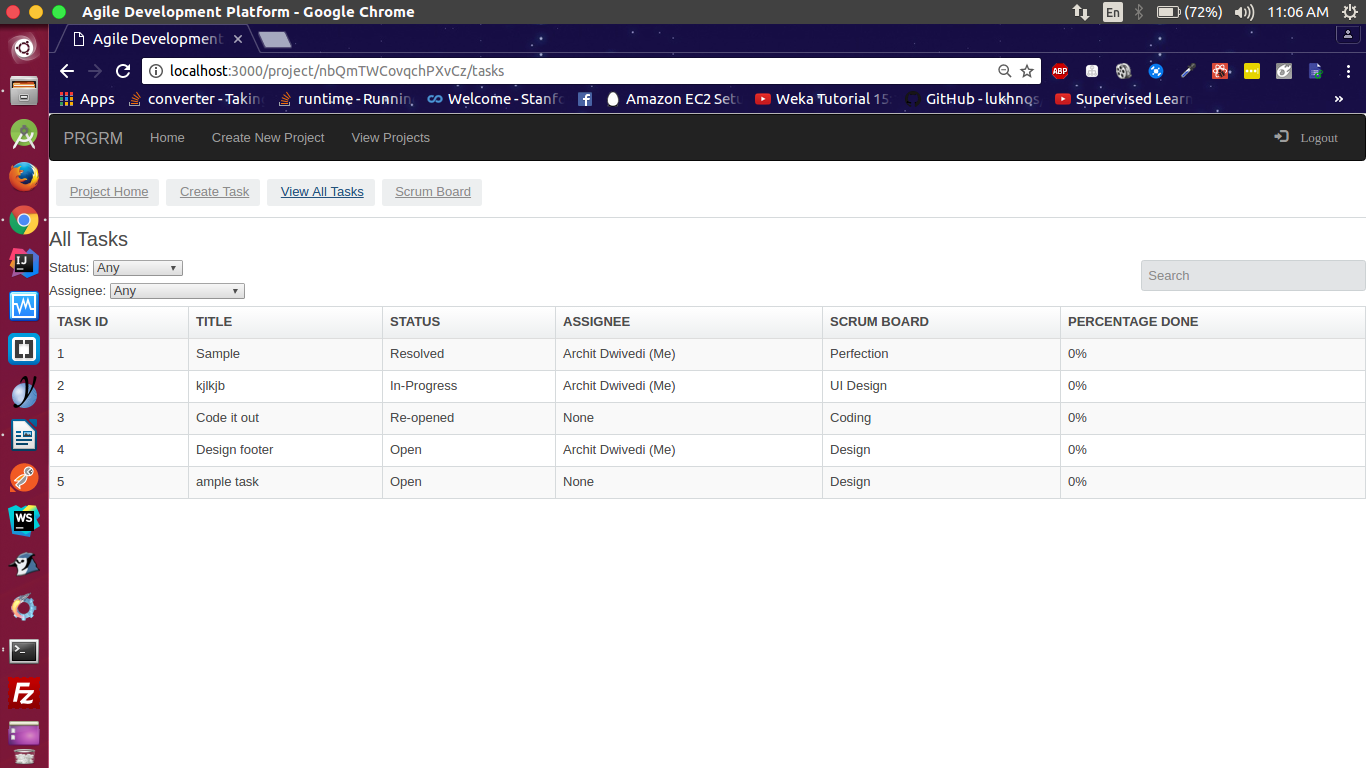
**5: CREATING A TASK**



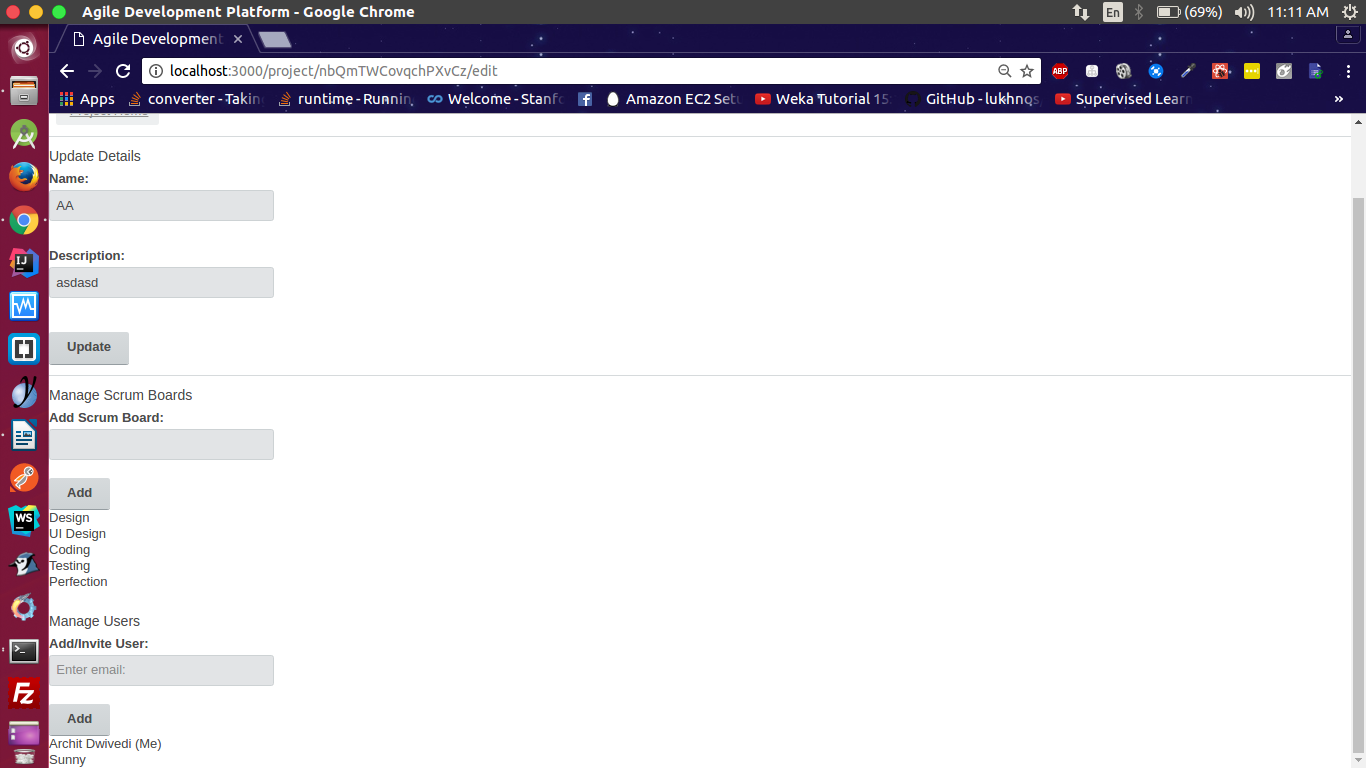
**6: SCRUM BOARD**



**7: VIEW ALL TASKS**



**8: SETTINGS PAGE**



**CHAPTER 9**

**SYSTEM SPECIFICATION**

**8.1 SOFTWARE REQUIREMENTS**

1. Android , Web App, Ios App, Desktop App
2. Internet Connectivity

**8.2 HARDWARE REQUIREMENTS:**

1. It would work on any platform due to CORDOVA.
2. Requires only 5 mb of ram at client end
3. 2 g.b of ram for SERVER (Back end of the app)

**CHAPTER 10**

# CONCLUSION

The efficiency of any system designed to suit an organization depends cooperation during the implementation stage and also flexibility of the system to adopt itself to the organization. “Net-Aucs” has been developed to overcome the problems with traditional Auction systems.

As evidence of the success of this mission, there are millions of items listed each day in thousands of different categories. There are items for almost any interest that one could imagine, from sheet music to automobiles to hand tools to real estate. And the variety doesn’t stop there. Need a computer? One may find it listed in the proper category, in any configuration from very old and obsolete to the latest greatest machine available. What about antiques? One can find an antique quilt that is up for highest bid, or maybe an old violin, whose beautiful tones have enchanted many though its years. Tickets. Maybe a ticket to the next concert of ones favorite artist or play production. One can even find that special bottle of wine, some aged, exotic cheese, and the perfect ‘mood’ music for that special occasion.

# Reliability

For a business of this nature, the need to be available is all-important. After all, when the only means of access is via ‘online’, it is of utmost importance to have and maintain an accessible web presence.

**CHAPTER 11**

**REFERENCES**

eBay, Sellers Guide, retrieved 11 April 2002 from:

http://pages.ebay.com/help/sellerguide/after-whatnow.html

eBay, Company Overview, retrieved 11 April 2002 from:

http://pages.ebay.com/community/aboutebay/overview/index.html)

(Mulrean, J., 5 steps to avoiding auction rip-offs,

retrieved 12 April 2002 from MSN Money,

http://money.msn.com/articles/smartbuy/dollarwise/9013.asping ground for fraud.)

http://www.newsfactor.com/perl/story/16387.html

<http://auctions.indiatimes.com/webapp/commerce/command/ExecMacro/TIL/macros/Logon.d2w/report>

https://www.baazee.com/jsp/UserRegistration.jsp

http://www.fabmart.com/HTML/faq.asp

http://www.baazee.com/static/help/User-agreement.html