Experiment No.: 1

Aim: Application of at least two traditional process models.

Software Used: MS Word

Theory: Software Processes is a coherent set of activities for specifying, designing, implementing, and testing software systems. A software process model is an abstract representation of a process that presents a description of a process from some perspective. There are many different software processes but all involve:

- Specification – defining what the system should do;
- Design and implementation – defining the organization of the system and implementing the system;
- Validation – checking that it does what the customer wants;
- Evolution – changing the system in response to changing customer needs.

Types of Software Process Model

Software processes, methodologies and frameworks range from specific prescriptive steps that can be used directly by an organization in day-to-day work, to flexible frameworks that an organization uses to generate a custom set of steps tailored to the needs of a specific project or group. In some cases, a "sponsor" or "maintenance" organization distributes an official set of documents that describe the process.

Software Process and Software Development Lifecycle Model

One of the basic notions of the software development process is SDLC models which stands for Software Development Life Cycle models. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out. The most used, popular and important SDLC models are given below:

- Waterfall model
- V model
- Incremental model
- RAD model
- Iterative model
- Spiral model
- Prototype model
- Agile model

Time Scheduler:

The process model which is used by Time scheduler project is known as Waterfall model. The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

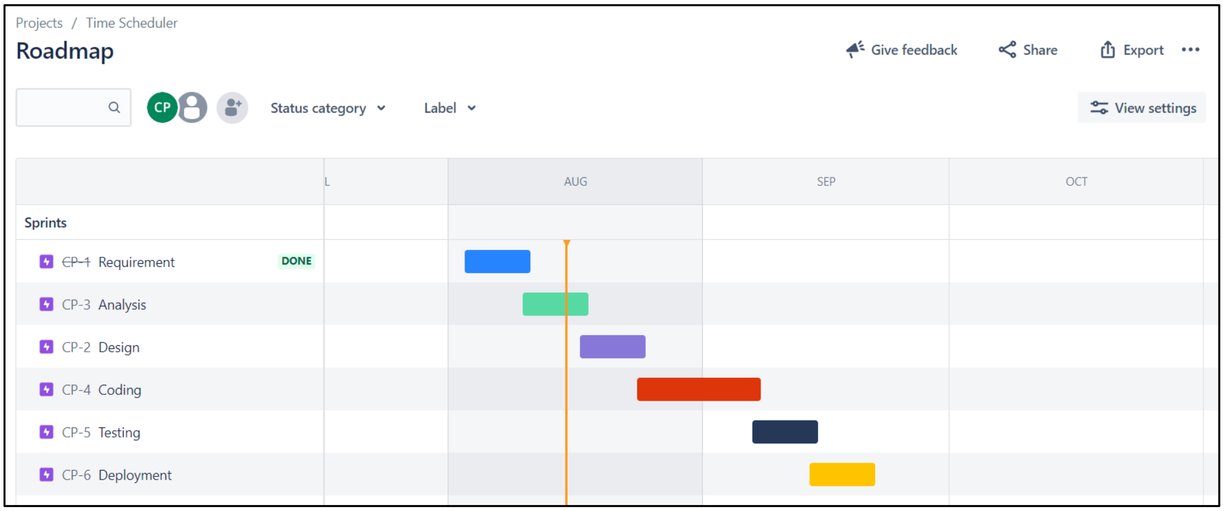
Situations where the use of Waterfall model is most appropriate are –

- Requirements are very well documented, clear, and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Some of the major advantages for the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Implementation:



~~CP-1~~ Requirement DONE

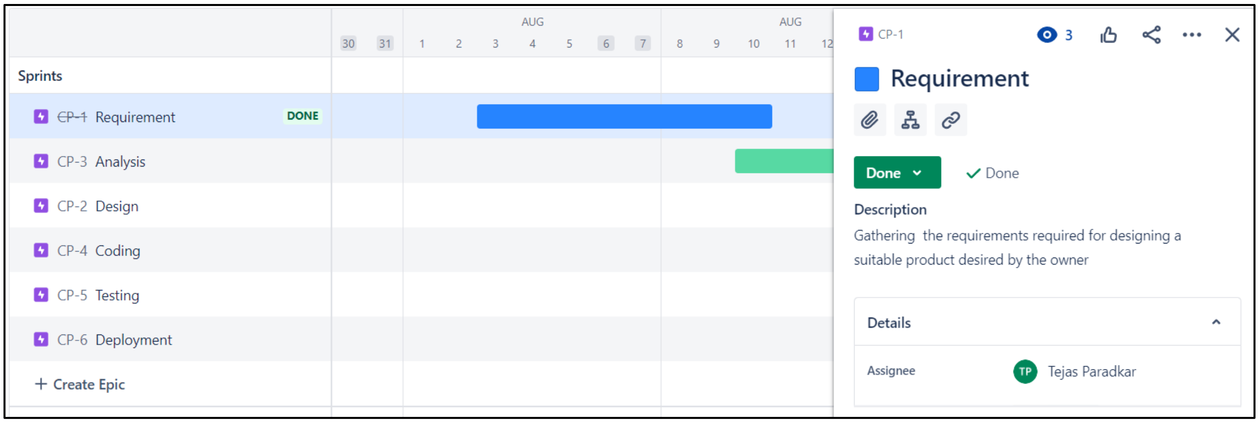
CP-3 Analysis

CP-2 Design

CP-4 Coding

CP-5 Testing

CP-6 Deployment



Sprints

~~CP-1~~ Requirement DONE

CP-3 Analysis

CP-2 Design

CP-4 Coding

CP-5 Testing

CP-6 Deployment

+ Create Epic

CP-1 3

Requirement

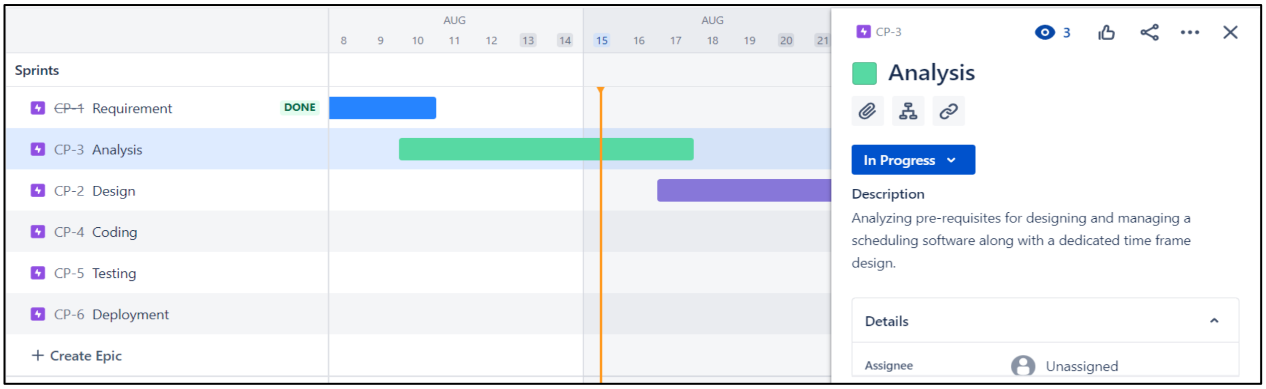
Done ✓ Done

Description

Gathering the requirements required for designing a suitable product desired by the owner

Details

Assignee TP Tejas Paradkar



Sprints

~~CP-1~~ Requirement DONE

CP-3 Analysis

CP-2 Design

CP-4 Coding

CP-5 Testing

CP-6 Deployment

+ Create Epic

CP-3 3

Analysis

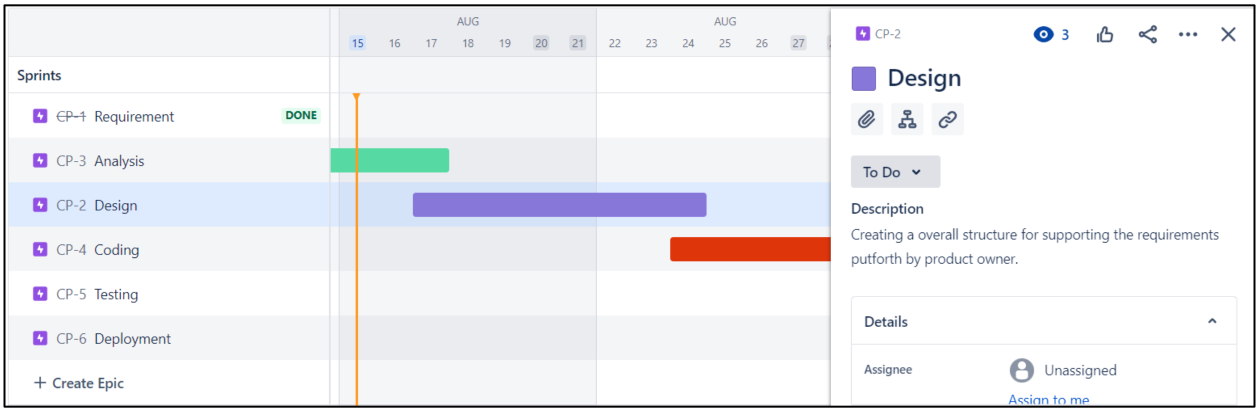
In Progress

Description

Analyzing pre-requisites for designing and managing a scheduling software along with a dedicated time frame design.

Details

Assignee Unassigned



Sprints

~~CP-1~~ Requirement DONE

CP-3 Analysis

CP-2 Design

CP-4 Coding

CP-5 Testing

CP-6 Deployment

+ Create Epic

CP-2 3

Design

To Do

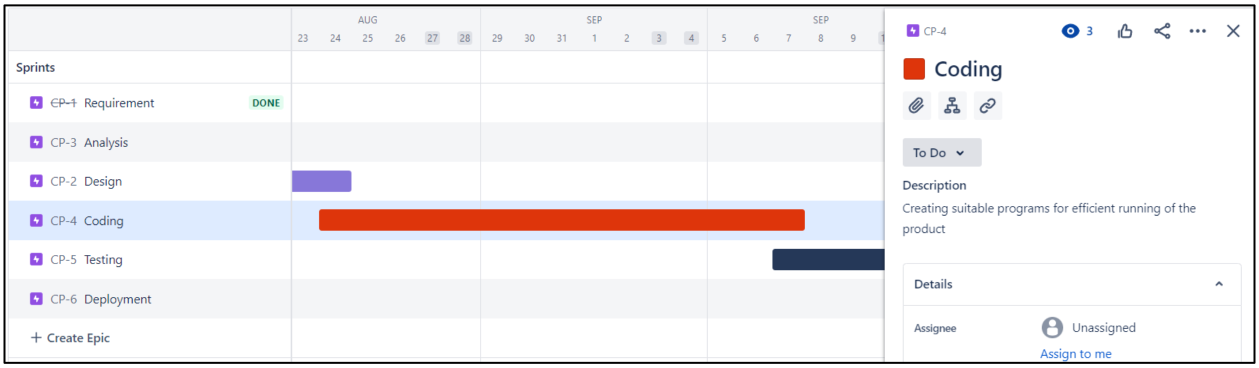
Description

Creating a overall structure for supporting the requirements putforth by product owner.

Details

Assignee Unassigned

Assign to me



Sprints

~~CP-1~~ Requirement DONE

CP-3 Analysis

CP-2 Design

CP-4 Coding

CP-5 Testing

CP-6 Deployment

+ Create Epic

CP-4 3

Coding

To Do

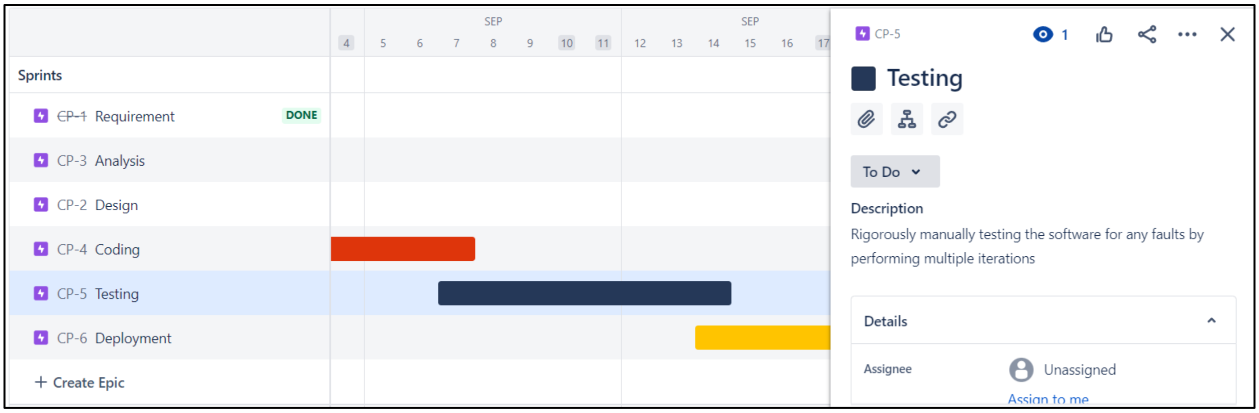
Description

Creating suitable programs for efficient running of the product

Details

Assignee Unassigned

Assign to me



Sprints

- ~~CP-1~~ Requirement DONE
- CP-3 Analysis
- CP-2 Design
- CP-4 Coding
- CP-5 Testing
- CP-6 Deployment

CP-5

Testing

To Do

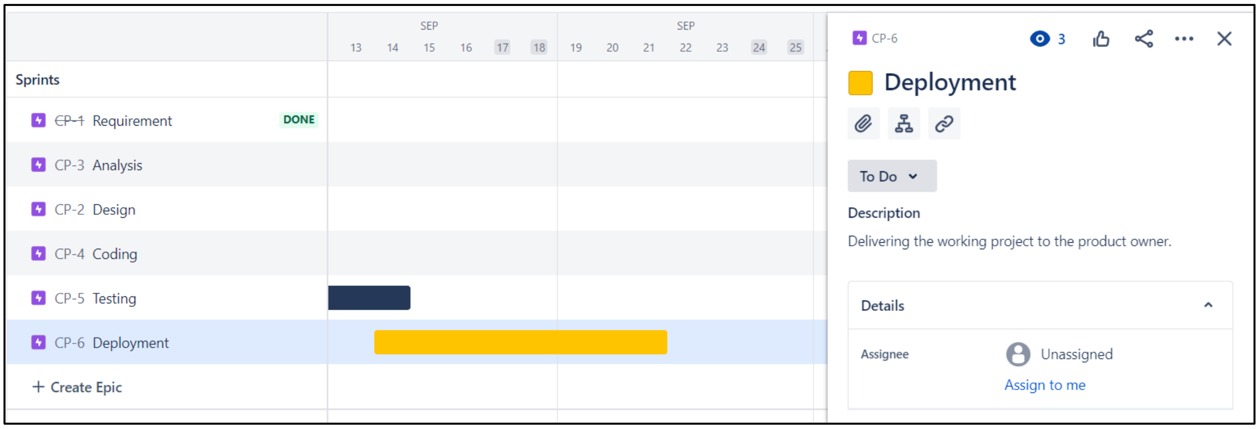
Description

Rigorously manually testing the software for any faults by performing multiple iterations

Details

Assignee Unassigned

Assign to me



Sprints

- ~~CP-1~~ Requirement DONE
- CP-3 Analysis
- CP-2 Design
- CP-4 Coding
- CP-5 Testing
- CP-6 Deployment

CP-6

Deployment

To Do

Description

Delivering the working project to the product owner.

Details

Assignee Unassigned

Assign to me

QR based Attendance:

The process model that is used by QR based Attendance project is known as RAD (Rapid Application Development) Model. The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

RAD model can be applied successfully to the projects in which clear modularization is possible. If the project cannot be broken into modules, RAD may fail.

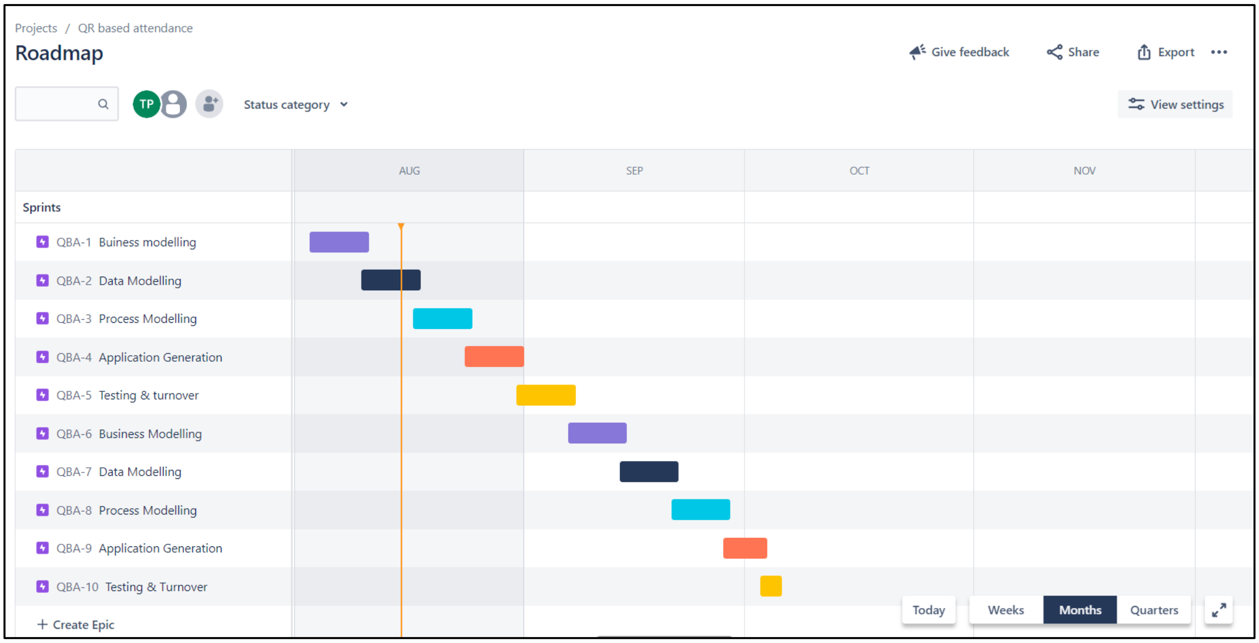
The applications of RAD are as follows –

- RAD should be used only when a system can be modularized to be delivered in an incremental manner.
- It should be used if there is a high availability of designers for Modelling.
- It should be used only if the budget permits use of automated code generating tools.
- RAD SDLC model should be chosen only if domain experts are available with relevant business knowledge.

The advantages of the RAD Model are as follows –

- Changing requirements can be accommodated.
- Progress can be measured.
- Iteration time can be short with use of powerful RAD tools.
- Productivity with fewer people in a short time.
- Reduced development time.
- Increases reusability of components.
- Quick initial reviews occur.
- Encourages customer feedback.
- Integration from very beginning solves a lot of integration issues.

Implementation:



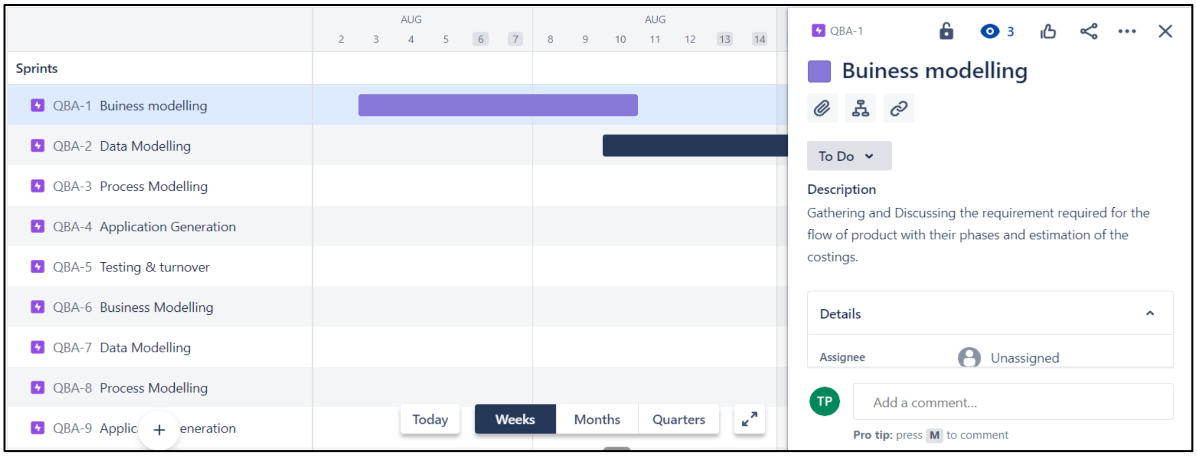
Projects / QR based attendance

Roadmap

Sprints

- QBA-1 Buiness modelling
- QBA-2 Data Modelling
- QBA-3 Process Modelling
- QBA-4 Application Generation
- QBA-5 Testing & turnover
- QBA-6 Business Modelling
- QBA-7 Data Modelling
- QBA-8 Process Modelling
- QBA-9 Application Generation
- QBA-10 Testing & Turnover

First term of the project runs from violet to yellow i.e., Business modelling to Testing & turnover. If client demands for more changes or improvements the whole term is repeated from start to end. These iterations can increase but will be limited to span of 60-90 days.



Sprints

QBA-1 Buiness modelling
QBA-2 Data Modelling
QBA-3 Process Modelling
QBA-4 Application Generation
QBA-5 Testing & turnover
QBA-6 Business Modelling
QBA-7 Data Modelling
QBA-8 Process Modelling
QBA-9 Applic eneration

QBA-1

Buiness modelling

To Do

Description

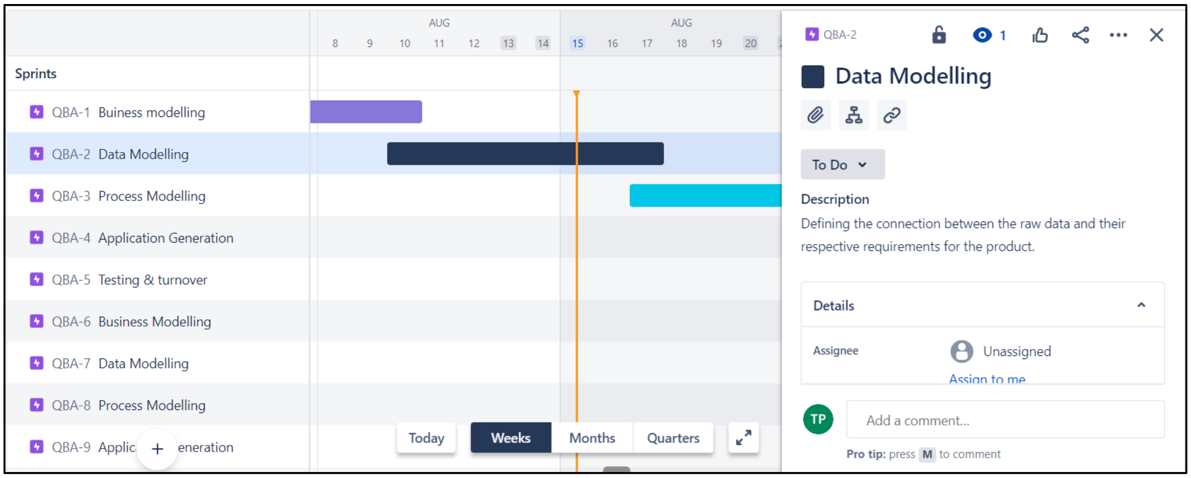
Gathering and Discussing the requirement required for the flow of product with their phases and estimation of the costings.

Details

Assignee Unassigned

Add a comment...

Pro tip: press M to comment



Sprints

QBA-1 Buiness modelling
QBA-2 Data Modelling
QBA-3 Process Modelling
QBA-4 Application Generation
QBA-5 Testing & turnover
QBA-6 Business Modelling
QBA-7 Data Modelling
QBA-8 Process Modelling
QBA-9 Applic eneration

QBA-2

Data Modelling

To Do

Description

Defining the connection between the raw data and their respective requirements for the product.

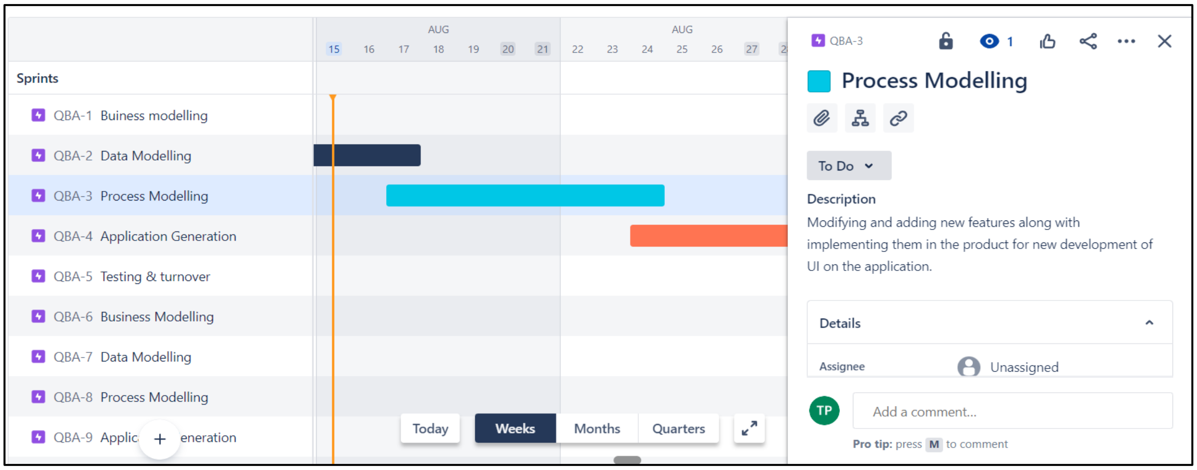
Details

Assignee Unassigned

Assign to me

Add a comment...

Pro tip: press M to comment



Sprints

QBA-1 Buiness modelling
QBA-2 Data Modelling
QBA-3 Process Modelling
QBA-4 Application Generation
QBA-5 Testing & turnover
QBA-6 Business Modelling
QBA-7 Data Modelling
QBA-8 Process Modelling
QBA-9 Applic eneration

QBA-3

Process Modelling

To Do

Description

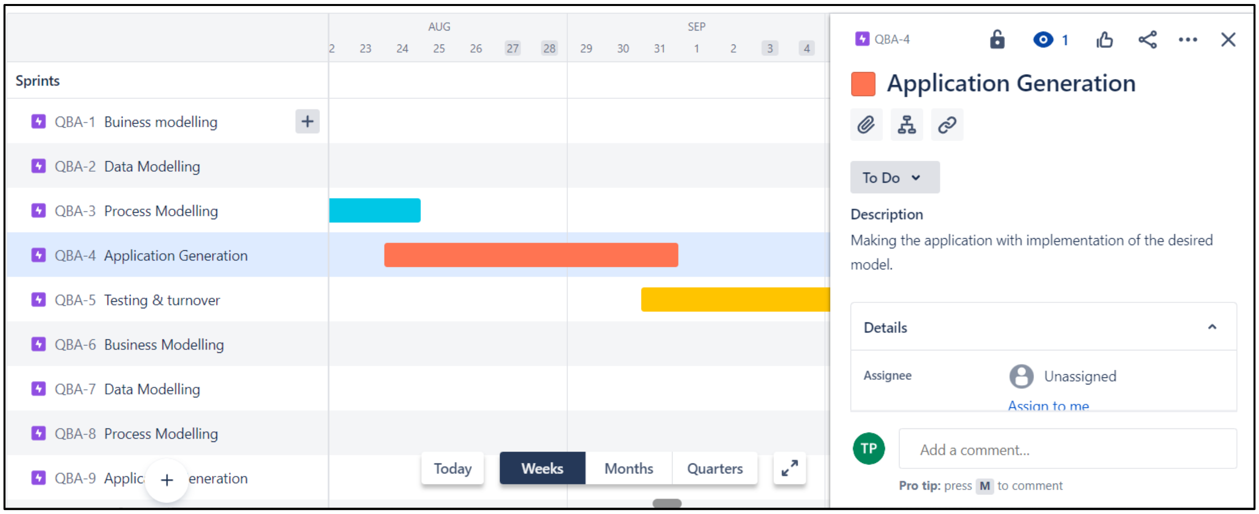
Modifying and adding new features along with implementing them in the product for new development of UI on the application.

Details

Assignee Unassigned

Add a comment...

Pro tip: press M to comment



Sprints

- QBA-1 Buiness modelling
- QBA-2 Data Modelling
- QBA-3 Process Modelling
- QBA-4 Application Generation
- QBA-5 Testing & turnover
- QBA-6 Business Modelling
- QBA-7 Data Modelling
- QBA-8 Process Modelling
- QBA-9 Applic + eneration

QBA-4

Application Generation

To Do

Description

Making the application with implementation of the desired model.

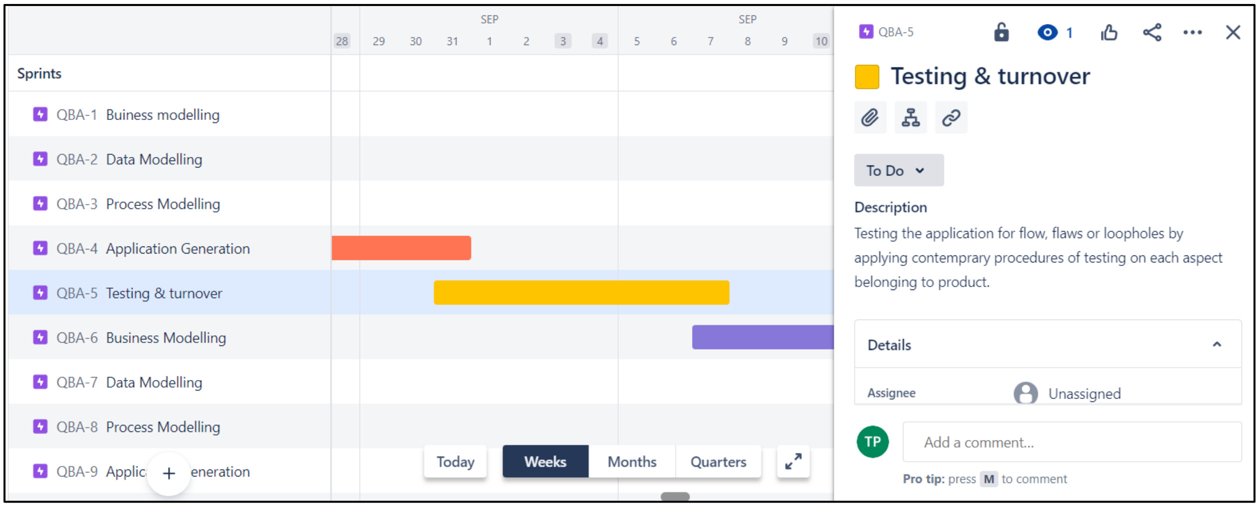
Details

Assignee Unassigned

Assign to me

TP Add a comment...

Pro tip: press M to comment



Sprints

- QBA-1 Buiness modelling
- QBA-2 Data Modelling
- QBA-3 Process Modelling
- QBA-4 Application Generation
- QBA-5 Testing & turnover
- QBA-6 Business Modelling
- QBA-7 Data Modelling
- QBA-8 Process Modelling
- QBA-9 Applic + eneration

QBA-5

Testing & turnover

To Do

Description

Testing the application for flow, flaws or loopholes by applying contemprary procedures of testing on each aspect belonging to product.

Details

Assignee Unassigned

TP Add a comment...

Pro tip: press M to comment

Conclusion: Thus, applied at least two traditional process models on multiple projects.

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

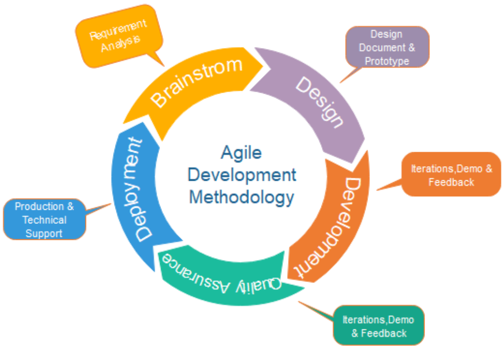
Experiment No: 2

Aim: Application of the Agile process models.

Software Used: JIRA

Theory: The meaning of Agile is swift or versatile. "**Agile process model**" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.


Requirement Analysis
Brainstrom
Design
Design Document & Prototype
Agile Development Methodology
Iterations,Demo & Feedback
Development
Production & Technical Support
Deployment
Quality Assurance
Iterations,Demo & Feedback

Every iteration involves cross functional teams working simultaneously on various areas like:

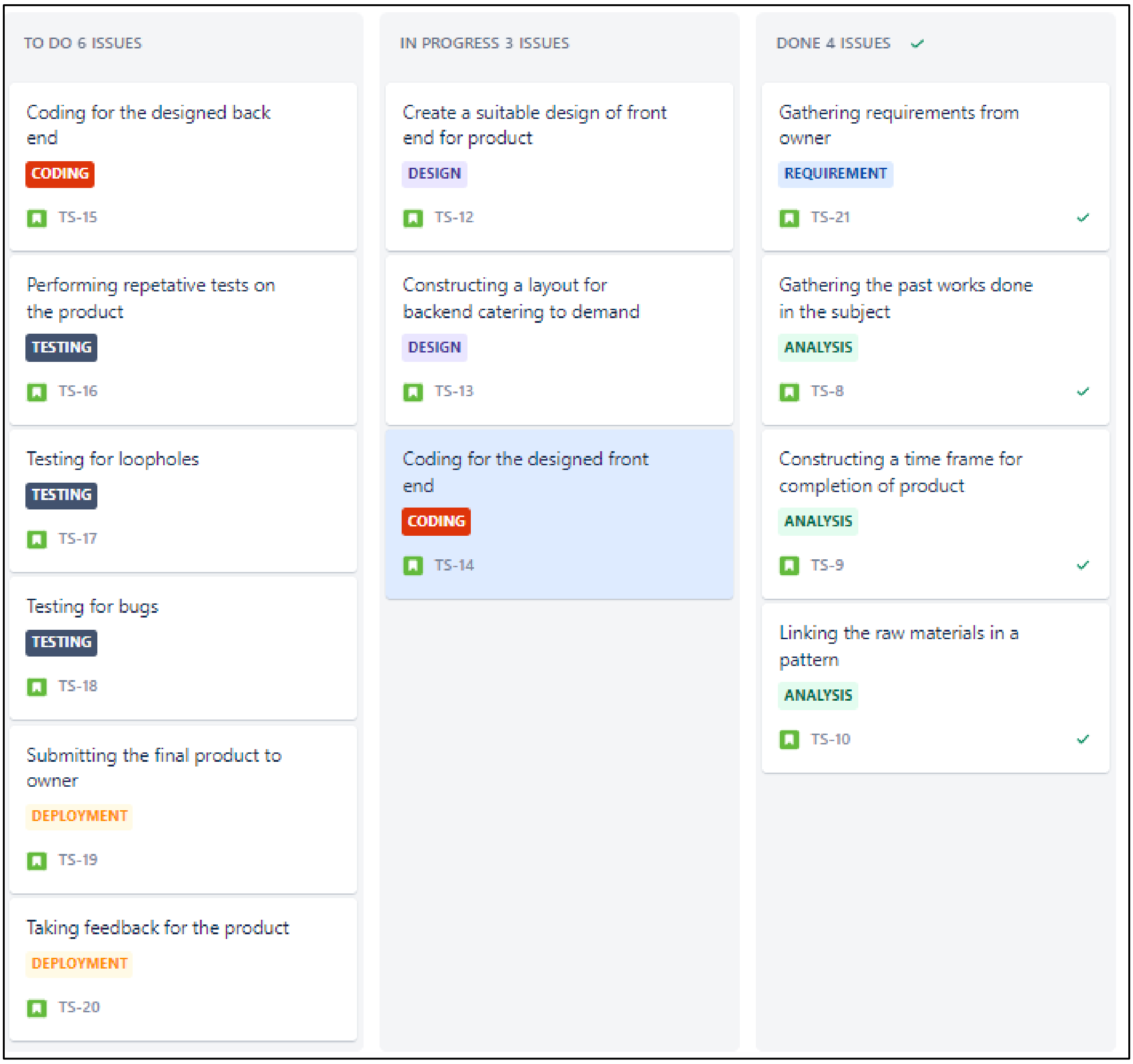
- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

Kanban methodology is based on the idea of continuous releases. Work is tracked using a kanban board that displays the statuses of work in columns and lanes. There are four important pillars to kanban to help teams ship products: continuous releases, WIP (work in progress) limits, the list of work, and columns or lanes. Here are some tools that come out-of-the-box in Jira Software's kanban template to help you run kanban with your team.

Implementation:

Kanban board for Time scheduler project.



TO DO 6 ISSUES

Coding for the designed back end
CODING
TS-15

Performing repetative tests on the product
TESTING
TS-16

Testing for loopholes
TESTING
TS-17

Testing for bugs
TESTING
TS-18

Submitting the final product to owner
DEPLOYMENT
TS-19

Taking feedback for the product
DEPLOYMENT
TS-20

IN PROGRESS 3 ISSUES

Create a suitable design of front end for product
DESIGN
TS-12

Constructing a layout for backend catering to demand
DESIGN
TS-13

Coding for the designed front end
CODING
TS-14

DONE 4 ISSUES ✓

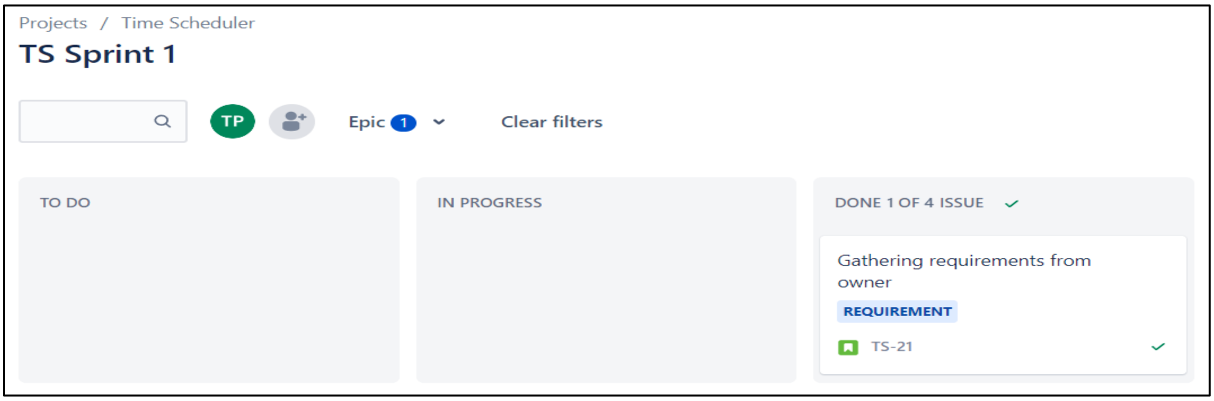
Gathering requirements from owner
REQUIREMENT
TS-21 ✓

Gathering the past works done in the subject
ANALYSIS
TS-8 ✓

Constructing a time frame for completion of product
ANALYSIS
TS-9 ✓

Linking the raw materials in a pattern
ANALYSIS
TS-10 ✓

Epic wise implementation on Kanban board.



Projects / Time Scheduler

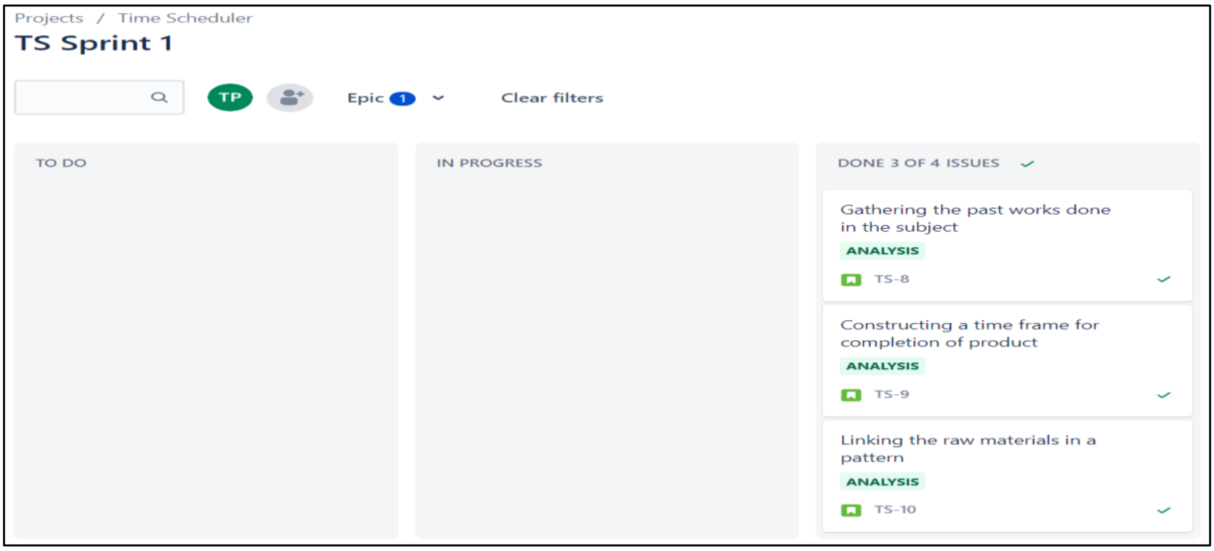
TS Sprint 1

TO DO

IN PROGRESS

DONE 1 OF 4 ISSUE ✓

Gathering requirements from owner
REQUIREMENT
TS-21 ✓



Projects / Time Scheduler

TS Sprint 1

TO DO

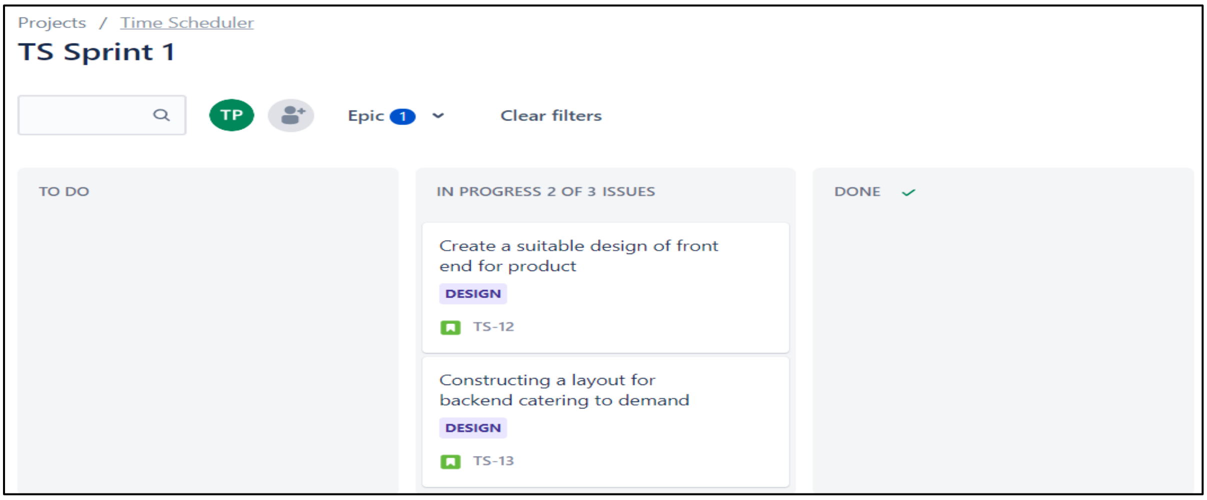
IN PROGRESS

DONE 3 OF 4 ISSUES ✓

Gathering the past works done in the subject
ANALYSIS
TS-8 ✓

Constructing a time frame for completion of product
ANALYSIS
TS-9 ✓

Linking the raw materials in a pattern
ANALYSIS
TS-10 ✓



Projects / Time Scheduler

TS Sprint 1

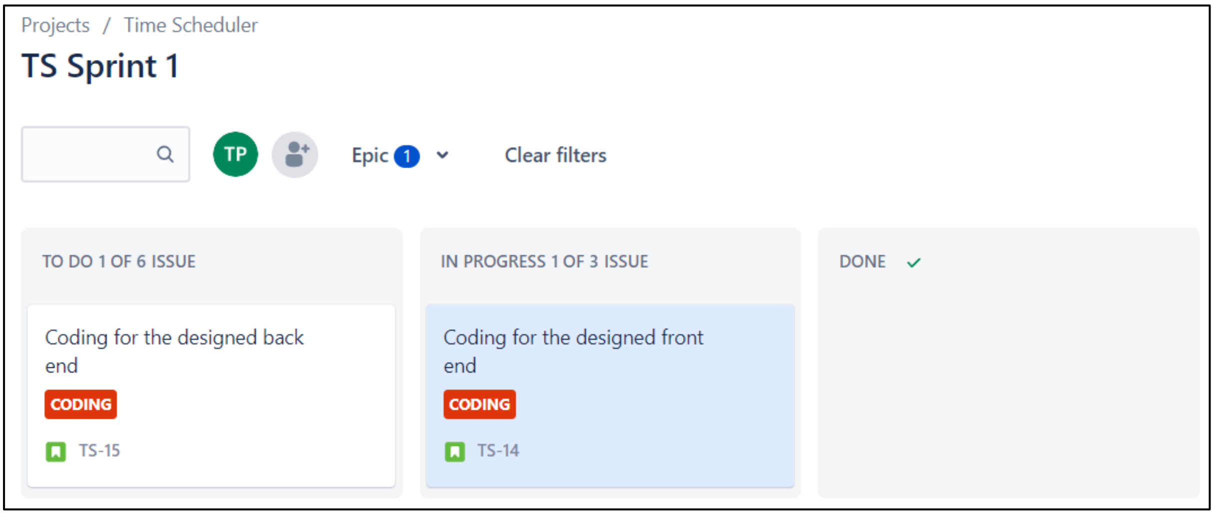
TO DO

IN PROGRESS 2 OF 3 ISSUES

Create a suitable design of front end for product
DESIGN
TS-12

Constructing a layout for backend catering to demand
DESIGN
TS-13

DONE ✓



Projects / Time Scheduler

TS Sprint 1

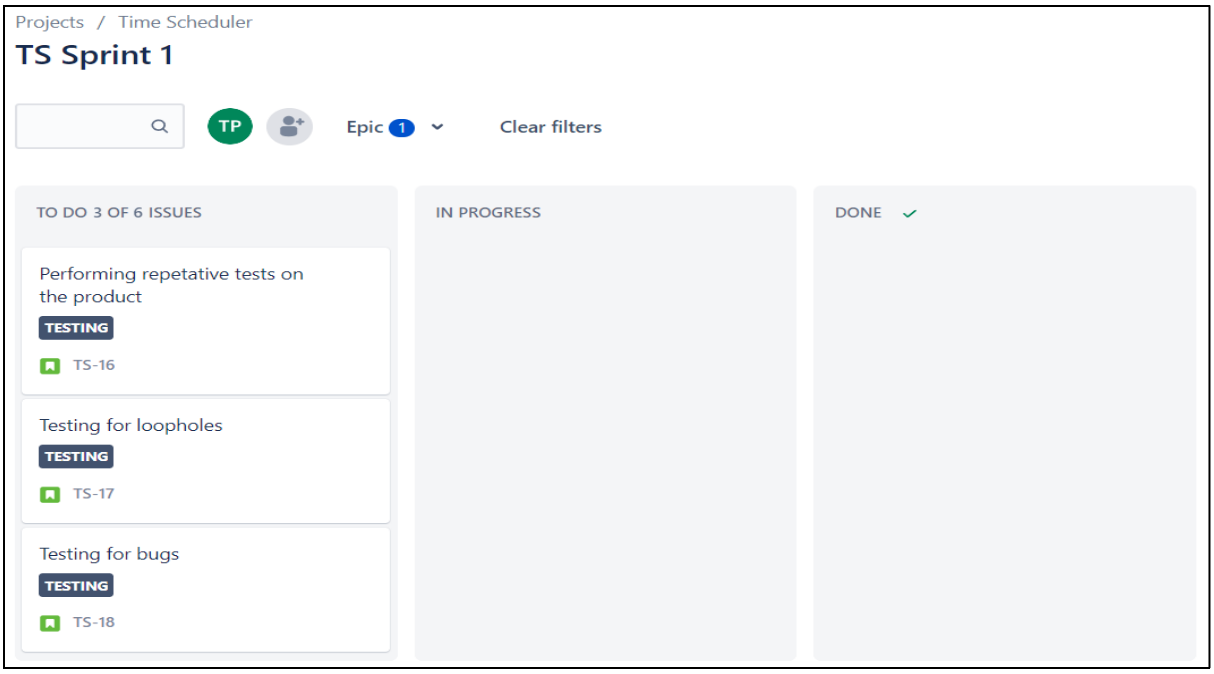
TO DO 1 OF 6 ISSUE

Coding for the designed back end
CODING
TS-15

IN PROGRESS 1 OF 3 ISSUE

Coding for the designed front end
CODING
TS-14

DONE ✓



Projects / Time Scheduler

TS Sprint 1

TP Epic 1 Clear filters

TO DO 3 OF 6 ISSUES

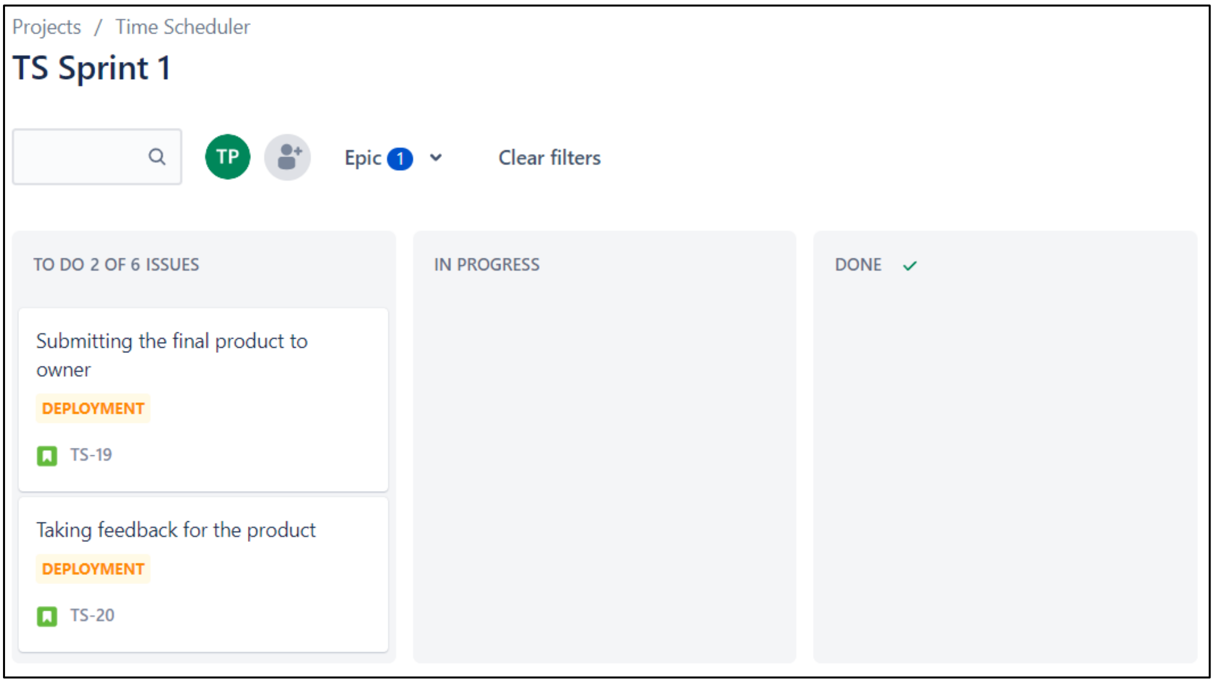
Performing repetative tests on the product
TESTING
TS-16

Testing for loopholes
TESTING
TS-17

Testing for bugs
TESTING
TS-18

IN PROGRESS

DONE ✓



Projects / Time Scheduler

TS Sprint 1

TP Epic 1 Clear filters

TO DO 2 OF 6 ISSUES

Submitting the final product to owner
DEPLOYMENT
TS-19

Taking feedback for the product
DEPLOYMENT
TS-20

IN PROGRESS

DONE ✓

Conclusion: Thus, applied Agile process models on our project.

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

SOFTWARE REQUIREMENTS SPECIFICATION
FOR
TIME SCHEDULER

PREPARED BY

TEJAS PARADKAR (B-65)
OMKAR PATIL (B-67)
KUNAL PATOLE (B-69)

GUIDE

PROF. SAUMYA SALIAN

Department of Computer Engineering

Datta Meghe College of Engineering

Airoli, Navi Mumbai-400708

University of Mumbai (AY 2021-22)

Table of Contents

1. INTRODUCTION

1.1) Purpose

In the existing android applications, we have generally had to set different remainders in the mobile phone for our schedule. We must set remainder again and again. It is a very time-consuming task for someone who does not have a fixed schedule or specific time. He/she must set remainder every time. Thus, the purpose behind the documentation is acquaint the intended audience behind the thought process of idea and possible upgrades in near future which also include multiple features such as a point reward system redeemable for the user after completing the required tasks along with upgrade within those.

1.2) Document Conventions:

1. All text contained in this document is in 12pt Times New Roman font.
2. Section titles are in 14pt Times New Roman and bold.
3. Subsections are in 14pt Time New Roman and underline.
4. Any further subsection breakdown is in 12pt Times New Roman.
5. All sections and sub-section are numbered using number format.
6. Document text will be single spaced and maintains a 1” margins.

1.3) Intended Audience:

The Time scheduler is app which is intended to audience which wants to cater his/her needs of sorting work by scheduling it in a particular timeframe by having organized slots for lectures as well as events he/she wants to attend. Teachers or employers are intended audience as well since they will be able to authenticate the authenticity of the program along with its timeframe and adjust workflow accordingly.

1.4) Product Scope:

The time scheduler app provides an interface between both employer and employee as well as teacher and student to track events or free times. Each user, teacher or student or event organizer will have different user interface. The teacher will have a traditional manager interface, with students having basic user interface. The teachers and event organizers will allow for entering of new events, lectures, deleting events or lectures. Uploading tasks and notifications. Students will be allowed to search different lectures and their schedules along with events. The system will also provide a history of previous tasks assigned or completed.

1.5) References

1. Paper presented in third information systems international conference by Ricky Michael.
2. Paper by Pink and Purple Group from Washington State University (Vancouver).

2. OVERALL DESCRIPTION

2.1) Product perspective:

The Time Scheduler system will be a new self-contained product. Communication will be done via Android Interface. This system is developed for reminding about lectures and events that can be used by students as well as teachers. In addition to this, it also includes facility to set remainder for any event.

The proposed system will be as follows:

- To provide the faculty members as well as students to keep track of the lectures to be conducted including their venue.
- To create a database in the system that automatically keeps track of lectures in a systematic and efficient way.
- To serve the purpose of both general and timetable remainder in one as the user need not open a separate application for different types of remainders.
- To provide a graphical user interface that is easy and interactive to use.
- To provide summarized view of timetable in tabular form and description of the events.

2.2) Product Functionality

i. The log-in interface

- The system will allow new users to create account for faculty, student, and organizers.
- The system will allow existing users to log in to their account
- General help information will be also provided.
- Password resetting feature will be provided

ii. The faculty interfaces

- The system will update from the database upon log on.
- The system will allow for management of student accounts for creation and deletion of users.
- System will allow for searching of current and previous tasks.

iii. Manage Task and Event Interface

iv. Event Organizer Interface

- Enlist of ongoing events.
- Enrolled students and faculties.

v. Student interfaces

- The system will update database upon login
- Students will be allowed to mark tasks as completed.
- Show events to be attended and enrolled.
- Allow for changing passwords.
- Know free time of faculties.

2.3) Users and Characteristics:

The intended users of the Time Scheduler will be faculties, students, and event organizers. The user will need not set a remainder every time for a fixed schedule. It has been basically deigned for a periodic series of events such as timetable schedule for the teacher as well as students. A database is stored for a particular time table which includes name of the faculty members teaching the subject, the venue of the subject and time of lectures for subject including the duration of lectures as well as event. User can also set remainders for general purposes. The application is divided into two categories by General Remainder and Time Table remainder. The user can decide which remainder to use according to his/her requirement.

2.4) Operating Environment:

The time scheduler system will run on android platform. The system will support android versions up to 11. The system will need to communicate between android application along with SQL database software.

2.5) Design and Implementation Constraints:

The system must be programmed in an object-oriented language, in this case we will be using android with support of Java. The system must use an active internet connection or the user will not be able to communicate with database. The Time Scheduler system must be portable so that multiple connections may be user to look at the information. The SQL database must be attached to the system.

2.6) User Documentation:

There should be minimal help needed to run this system. Should there be any need for help, there will be help menu that will guide the user through using the system. This document will also serve as a documentation that specifies the detailed requirements and features of product.

2.7) Assumptions and Dependencies:

One assumption that could affect the design is that the user run a android operating system, which has a less version than 8. This can affect unknown effects on the system functionality. Another assumption that could affect the design is that the user will have adequate internet connection; this could affect the speed with which the interface communicates with the database. Lastly this system will be written for users with basic understanding of how mobile works. Users with less mobile experience may have a harder time.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1) User Interface:

The Time Scheduler system will have five main Graphical user Interfaces:

1. Log On:
   This interface will consist of login interface. It will contain text fields allowing user name and password for login with a corresponding button, a button for a help menu which will assist in usability and a button which will allow the manager to reset password. Depending on the type of user which logs on, either the faculty, event organizer or student interface will be loaded.

2. Faculty:
   This interface will consist of various tasks. This interface will allow for tasks search, password changes, event listings, notifications, and ability to navigate through management of tasks.

3. Event Organizer:
   This interface will consist of listed events, enrolled participants, event notifications and volunteers available as well as dates free.

4. Students:
   This interface will be limited version of parental interface. In this interface user will be able to join events, search tasks, mark tasks as completed, and change password.

3.2) Hardware Interface:

- The absolute minimum requirement for android were originally a 200 MHz processor, 3GB of RAM, 16GB of storage.
- For the base SDK package, at least 600MB of available disk space. For each platform downloaded into the SDK, an additional 100 MB is needed.
- Out of box, Android is incompatible with ARMv4 or lower: ARMv5 or higher is needed to run native code with modifications
- Android 4++ requires an ARMv7 processor. Custom Version of Android 4+ have been made for ARMv6 however.

3.3) Software Interface:

Start with the Android compatibility page. This outlines goals for Android’s compatibility and links to the current Compatibility Definition Document which has the technical requirements.

3.4) Communication Interfaces:

The Time Scheduler System will use the JDBC driver to communicate with SQL database.

4. SYSTEM FEATURES

System Features:

- Enables quick management of lectures timing & venue for both, the teachers, and students.
- Need not set the reminder repeatedly every time for each day.
- Stores the subject information like subject code, teacher's name, subject name, venue & schedule for the whole week.
- Quick adding, editing, deleting, and updating the event.
- Also helps to set the general reminders as that of the meetings and the other tasks.
- Gives the summarized view of the time table of the week & the description of the events to plan the timely schedule.
- User friendly & quick to use interface.

5. OTHER NON-FUNCTIONAL REQUIREMENTS

5.1) Performance Requirements:

The system shall perform basic operations in less than 2 seconds. While navigating the system, user interfaces should update quickly, this is not an issue and doesn't need to be addressed. The system shall run on a android platform. The system shall communicate between the android application and SQL database. The SQL database needs to have the capacity to grow.

5.2) Safety and Security Requirements:

The level of security for this product is refined mostly to the privacy needs between users. Improper entries in any profile may result in malicious activities leading to false pay-outs. Privacy between students is also important to the client because siblings, peers or other students using the product may utilize the ability to enter another student's profile with negative intent.

- Student account names will be associated with a password which will be chosen by the user upon first use.
- All passwords will be associated with one secret question to assist in password reset.
- If faculties enter the incorrect password three times, then they will be locked out.
- If student enter the incorrect password three times, then they will be locked out.

5.3) Software Quality Attributes:

1. Usability
   The interfaces of the system will be designed in a user-friendly manner such that the user will need to training to perform operations within the system. The interfaces will be easily navigable, clearly labelled, and help menu will be provided with instructions for performing basic tasks.

2. Maintainability
   The system code should be written to allow for future possible upgrades. Code will be documented, including version updates and authors. Code will be fully commented. Each method will include a description of its function and any additional information needed to help in future additions.

3. Portability
   The system is portable to operate it on a Android operating system; although effects of using previous versions of android are unknown at this time and require future testing.

5.4) Business Rules:

- A repository, allowing decision logic to be externalized from core application code.
- Tools, allowing both technical developers and business experts to define and manage decision logic.
- A runtime environment, allowing applications to invoke decision logic managed within the BRMS (Business Rule Management System) and execute it using a business rules engine
- Reduced or removed reliance on IT departments for changes in live systems. Although, QA and Rules testing would still be needed in any enterprise system.
- Increased control over implemented decision logic for compliance and better business management
- The ability to express decision logic with increased precision, using a business vocabulary syntax and graphical rule representations (decision tables, trees, scorecards, and flows)
- Improved efficiency of processes through increased decision automation.

Conclusion: Thus, prepared the SRS for project.

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

Experiment No: 4

Aim: Structured Data Flow Analysis

Software Used: Star UML

Theory: Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

It is a systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user.

It has following attributes −

- It is graphic which specifies the presentation of application.
- It divides the processes so that it gives a clear picture of system flow.
- It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
- It is an approach that works from high-level overviews to lower-level details.

Structured Analysis Tools

During Structured Analysis, various tools and techniques are used for system development. They are −

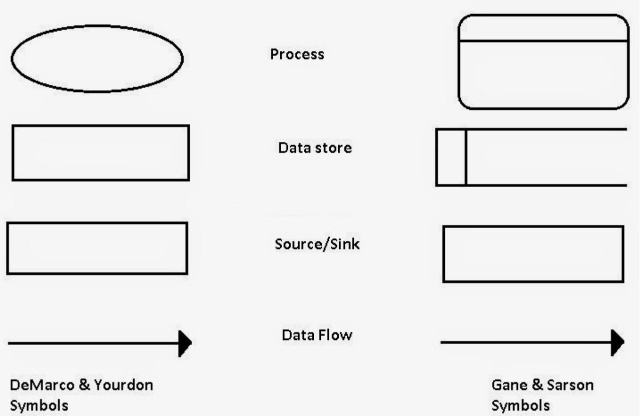
- Data Flow Diagrams
- Data Dictionary
- Decision Trees
- Decision Tables
- Structured English
- Pseudo code

Data Flow Diagrams (DFD)

It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

- It shows the flow of data between various functions of system and specifies how the current system is implemented.
- It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
- Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
- It gives an overview of what data a system processes, what transformations are performed, what data are stored, what results are produced and where they flow.

Symbols used in DFD



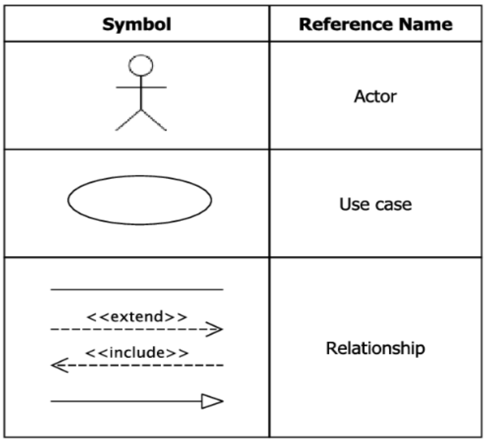
Used Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

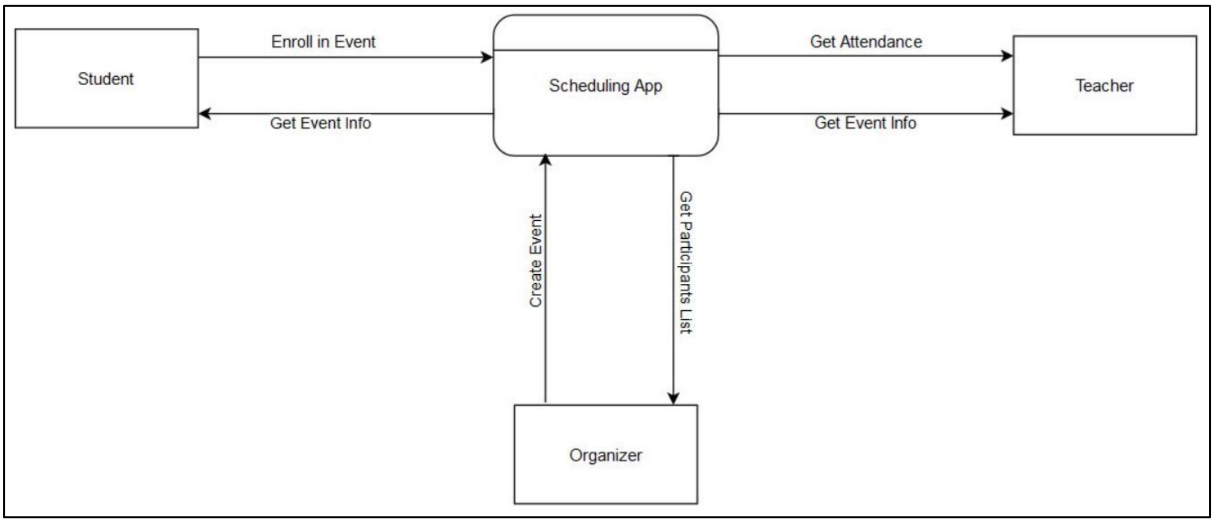
UML use case diagrams are ideal for:

- Representing the goals of system-user interactions
- Defining and organizing functional requirements in a system
- Specifying the context and requirements of a system
- Modeling the basic flow of events in a use case

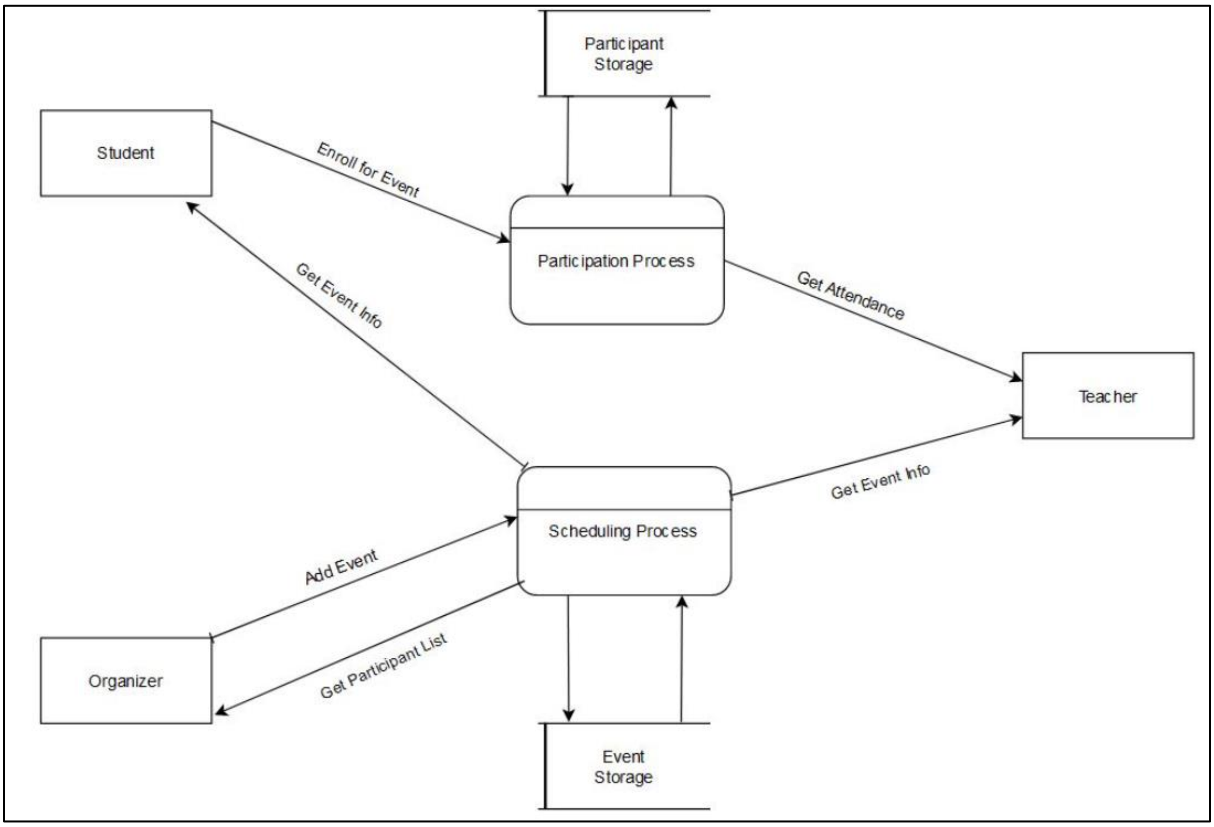


Symbol	Reference Name
	Actor
	Use case
<<extend>> <<include>>	Relationship

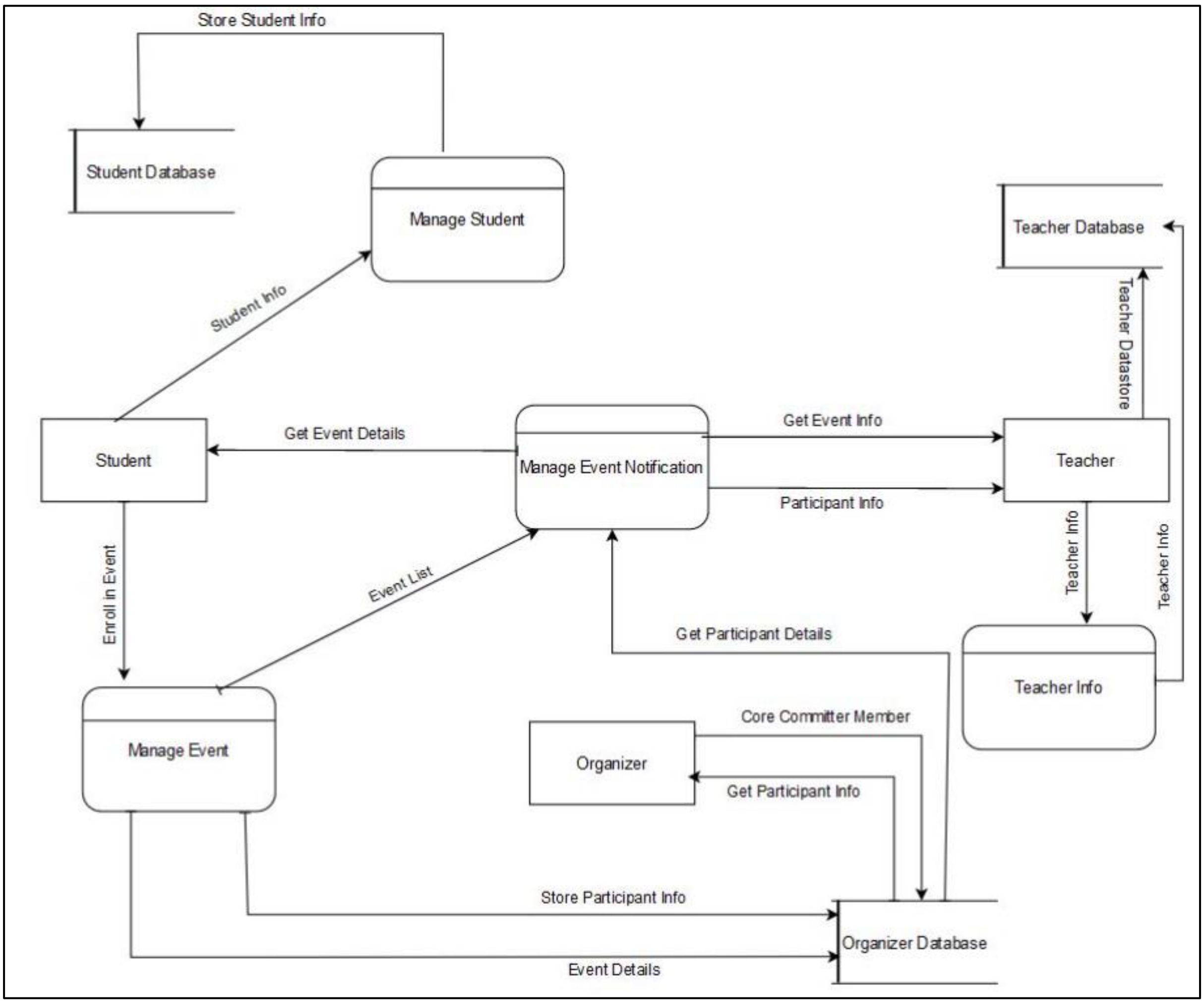
Level 0 DFD:



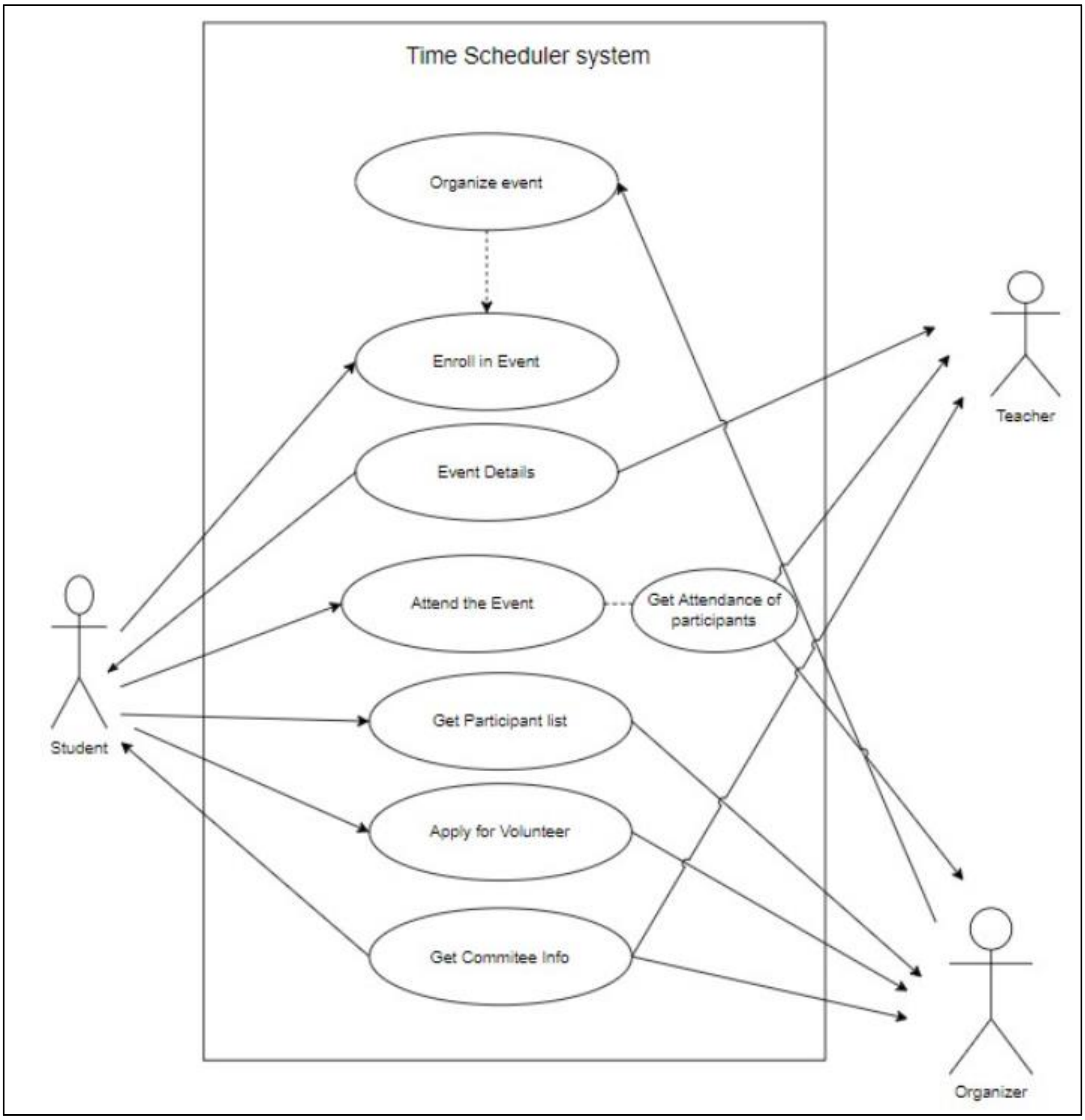
Level 1 DFD:



Level 2 DFD:


Store Student Info
Student Database
Manage Student
Teacher Database
Student Info
Teacher Datastore
Student
Get Event Details
Manage Event Notification
Get Event Info
Participant Info
Teacher
Enroll in Event
Event List
Teacher Info
Teacher Info
Get Participant Details
Teacher Info
Manage Event
Core Committer Member
Organizer
Get Participant Info
Store Participant Info
Organizer Database
Event Details

Use Case Diagram:



Conclusion: Thus, Structured Data Flow Analysis was performed on project.

Rubrics →	Timely Submission (5)	Neatness (5)	Knowledge (5)	Signature of staff With Date
Marks →				

Experiment No: 5

Aim: Use of Metrics to estimate the cost

Software Used: MS-Word

Software Cost Estimation

For any new software project, it is necessary to know how much it will cost to develop and how much development time will it take. These estimates are needed before development is initiated, but how is this done? Several estimation procedures have been developed and are having the following attributes in common.

1. Project scope must be established in advanced.
2. Software metrics are used as a support from which evaluation is made.
3. The project is broken into small PCs which are estimated individually. To achieve true cost & schedule estimate, several options arise.
4. Delay estimation
5. Used symbol decomposition techniques to generate project cost and schedule estimates.
6. Acquire one or more automated estimation tools.

Uses of Cost Estimation

1. During the planning stage, one needs to choose how many engineers are required for the project and to develop a schedule.
2. In monitoring the project's progress, one needs to access whether the project is progressing according to the procedure and takes corrective action, if necessary.
3. Estimation is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable.
4. Estimation determines how much money, effort, resources, and time it will take to build a specific system or product.

Loc-based Cost Estimation

The LOC (Line of Code) is a product size metric in software engineering. Here, the number of lines in the code are counted and based on the number of lines the cost is calculated.

LOC-based Estimation

- Different languages lead to different lengths of code
- It is not clear how to count lines of code
- A report, screen, or GUI generator, can generate thousands of lines of code in minutes

- Depending on the application, the complexity of code is different

Function Points: FP

Function Points is used in 2 contexts:

- **Past:** To develop **metrics** from historical data
- **Future:** Use of available metrics to size the s/w of a new project

FP-based Estimation

- Based on FP metric for the size of a product
- Based on the number of inputs (Inp), outputs (Out), inquiries (Inq), master files (Maf), interfaces (Inf)
- Classify each component of the product (Inp, Out, Inq, Maf, Inf) as simple, average, or complex (next slide)

Implementation

LOC-based Estimation

Function	**Estimated LOC**
User interface	5000
2-D geometric analysis	2000
3 D geometric analysis	2000
Database management	6500
Graphic display facilities	4000
I/O control function	2100
Analysis function	8400
Total estimated	**30000**

Average productivity based on historical data **10000** LOC/pm

Rs. 50000 per month/PP

Rs. 5/LOC

If the estimated project is **30000** LOC,
then the total estimated project cost is **Rs.1,50,000**(30000 x 5) and
the estimated effort is **3** (30000/10000) person-months

Measurement Parameters	Count	Weighting Factor	Total
		Average	
Number of user inputs	20	04	80
Number of user outputs	18	05	90
Number of user inquiries	10	04	40
Number of files	06	10	60
Number of external interfaces	10	07	70
Complexity multiplier			03
Count total			**340**

$\sum(Fi) = 14 \text{ x } 3$

$\boldsymbol{\sum(Fi) = 42}$

Formula: $FP = \text{count total x } [0.65 + 0.01 \text{ x } \sum(Fi)]$

$FP = 340 \text{ x } \{0.65 + 0.01 \text{ x } 42\}$

$FP = 340 \text{ x } \{0.65 + 0.42\}$

$\mathbf{FP = 364}$

Therefore, the Function Point is 364.

Conclusion: Thus, Use of Metrics is done to estimate the cost of project.

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

Experiment No: 6

Aim: Scheduling and tracking of the project

Software Used: Gantt Project

Theory: -

Project Scheduling

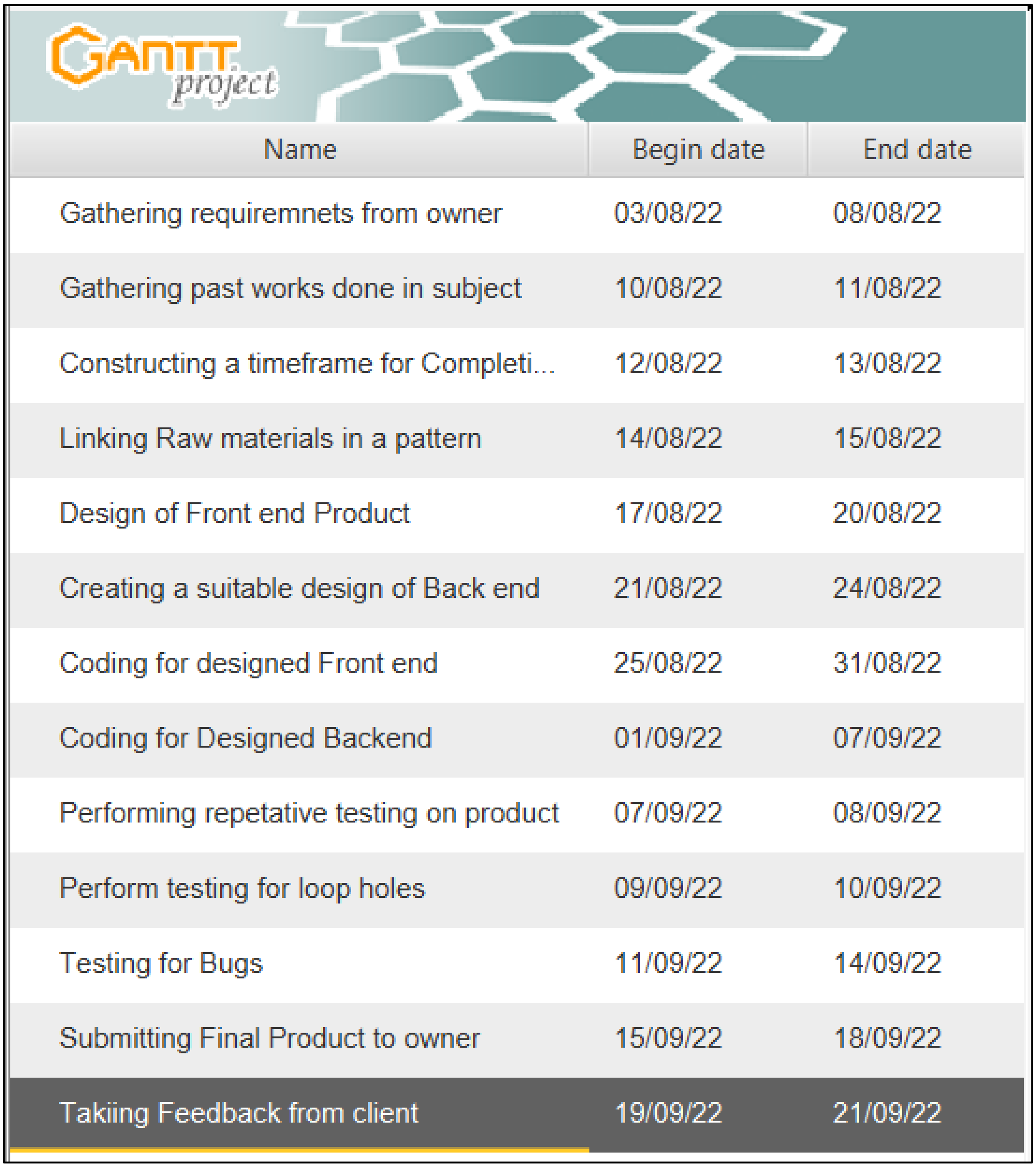
Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. To schedule the project plan, a software project manager wants to do the following:

1. Identify all the functions required to complete the project.
2. Break down large functions into small activities.
3. Determine the dependency among various activities.
4. Establish the most likely size for the time duration required to complete the activities.
5. Allocate resources to activities.
6. Plan the beginning and ending dates for different activities.
7. Determine the critical path. A critical way is the group of activities that decide the duration of the project.

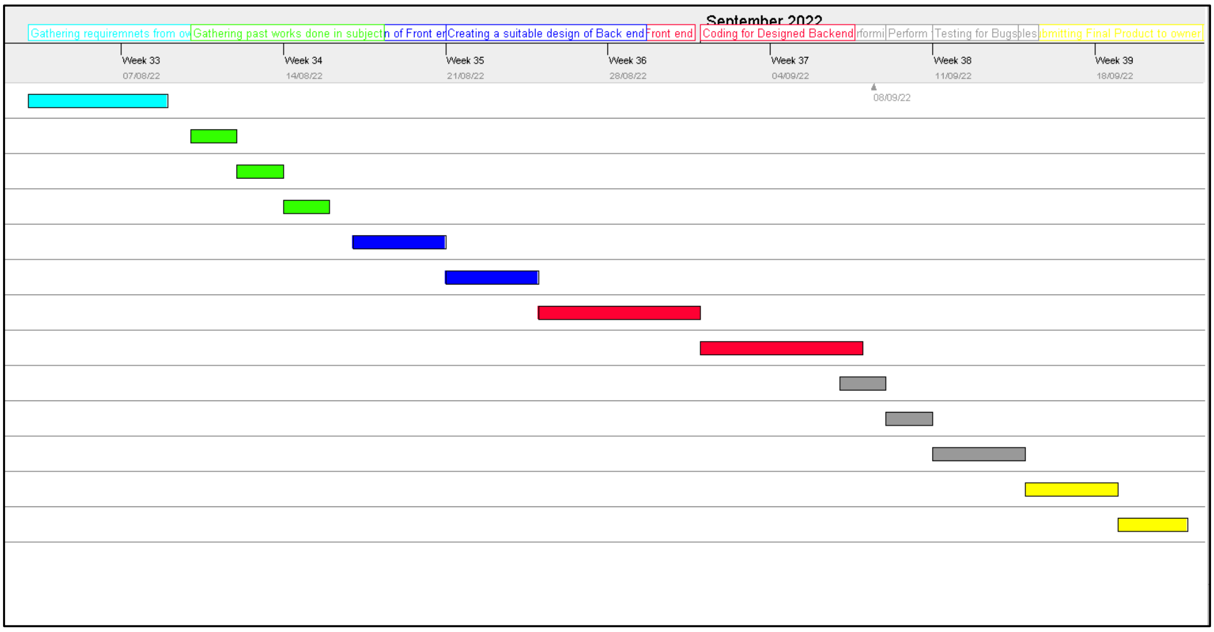
The first method in scheduling a software plan involves identifying all the functions required to complete the project. A good judgment of the intricacies of the project and the development process helps the supervisor to identify the critical role of the project effectively. Next, the large functions are broken down into a valid set of small activities which would be assigned to various engineers. The work breakdown structure formalism supports the manager to breakdown the function systematically after the project manager has broken down the purpose and constructs the work breakdown structure; he has to find the dependency among the activities. Dependency among the various activities determines the order in which the various events would be carried out. If an activity A necessary the results of another activity B, then activity A must be scheduled after activity B. In general, the function dependencies describe a partial ordering among functions, i.e., each service may precede a subset of other functions, but some functions might not have any precedence ordering describe between them (called concurrent function). The dependency among the activities is defined in the pattern of an activity network.

Once the activity network representation has been processed out, resources are allocated to every activity. Resource allocation is usually done using a Gantt chart. After resource allocation is completed, a PERT chart representation is developed. The PERT chart representation is useful for program monitoring and control. For task scheduling, the project plan needs to decompose the project functions into a set of activities. The time frame when every activity is to be performed is to be determined. The end of every action is called a milestone. The project manager tracks the function of a project by audit the timely completion of the milestones. If he examines that the milestones start getting delayed, then he has to handle the activities carefully so that the complete deadline can still be met.

Implementation:



Name	Begin date	End date
Gathering requiremnets from owner	03/08/22	08/08/22
Gathering past works done in subject	10/08/22	11/08/22
Constructing a timeframe for Completi...	12/08/22	13/08/22
Linking Raw materials in a pattern	14/08/22	15/08/22
Design of Front end Product	17/08/22	20/08/22
Creating a suitable design of Back end	21/08/22	24/08/22
Coding for designed Front end	25/08/22	31/08/22
Coding for Designed Backend	01/09/22	07/09/22
Performing repetative testing on product	07/09/22	08/09/22
Perform testing for loop holes	09/09/22	10/09/22
Testing for Bugs	11/09/22	14/09/22
Submitting Final Product to owner	15/09/22	18/09/22
Takiing Feedback from client	19/09/22	21/09/22



Conclusion: Thus, performed scheduling and tracking of the project.

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

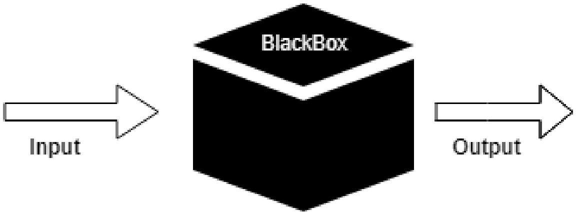
Experiment No: 7

Aim: Write test cases for black box testing

Software Used: Selenium/GitHub/Jira

Theory: -

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.
In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.



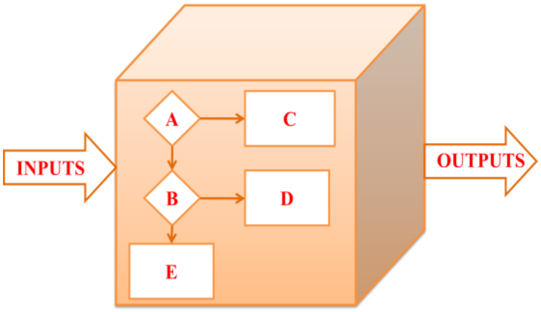
Generic steps of black box testing

- The black box test is based on the specification of requirements, so it is examined in the beginning.
- In the second step, the tester creates a positive test scenario and an adverse test scenario by selecting valid and invalid input values to check that the software is processing them correctly or incorrectly.
- In the third step, the tester develops various test cases such as decision table, all pairs test, equivalent division, error estimation, cause-effect graph, etc.
- The fourth phase includes the execution of all test cases.
- In the fifth step, the tester compares the expected output against the actual output.
- In the sixth and final step, if there is any flaw in the software, then it is cured and tested again.

White Box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

The developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.



Test procedure

The test procedure of black box testing is a kind of process in which the tester has specific knowledge about the software's work, and it develops test cases to check the accuracy of the software's functionality.

It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function. A tester knows about the definite output of a particular input, but not about how the result is arising. There are various techniques used in black box testing for testing like decision table technique, boundary value analysis technique, state transition, All-pair testing, cause-effect graph technique, equivalence partitioning technique, error guessing technique, use case technique and user story technique.

Test cases

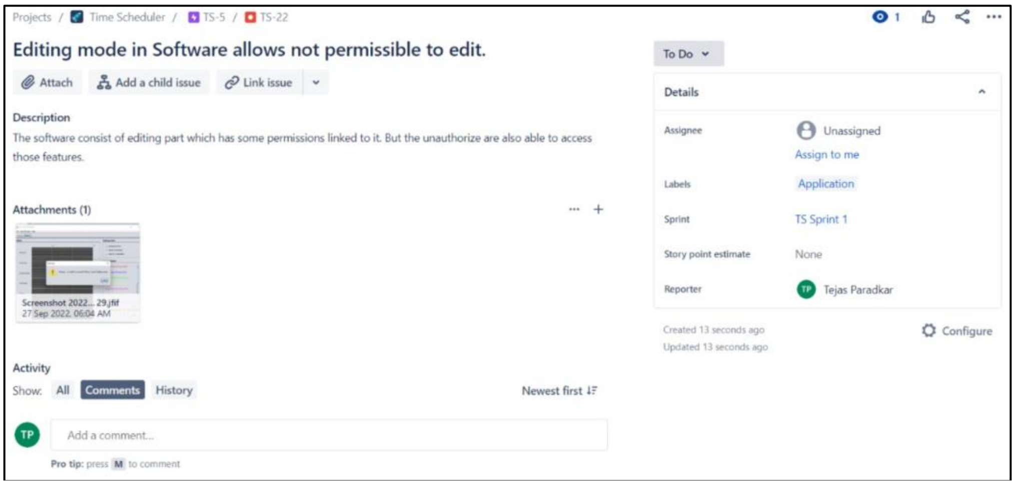
Test cases are created considering the specification of the requirements. These test cases are generally created from working descriptions of the software including requirements, design parameters, and other specifications. For the testing, the test designer selects both positive test scenario by taking valid input values and adverse test scenario by taking invalid input values to determine the correct output. Test cases are mainly designed for functional testing but can also be used for non-functional testing. Test cases are designed by the testing team; there is not any involvement of the development team of software.

Techniques Used in Black Box Testing

Decision Table Technique	Decision Table Technique is a systematic approach where various input combinations and their respective system behavior are captured in a tabular form. It is appropriate for the functions that have a logical relationship between two and more than two inputs.
Boundary Value	Boundary Value Technique is used to test boundary values, boundary values are those that contain the upper and lower limit of a variable. It

Technique	tests, while entering boundary value whether the software is producing correct output or not.
State Transition Technique	State Transition Technique is used to capture the behavior of the software application when different input values are given to the same function. This applies to those types of applications that provide the specific number of attempts to access the application.
All-pair Testing Technique	All-pair testing Technique is used to test all the possible discrete combinations of values. This combinational method is used for testing the application that uses checkbox input, radio button input, list box, text box, etc.
Cause-Effect Technique	Cause-Effect Technique underlines the relationship between a given result and all the factors affecting the result. It is based on a collection of requirements.
Equivalence Partitioning Technique	Equivalence partitioning is a technique of software testing in which input data divided into partitions of valid and invalid values, and it is mandatory that all partitions must exhibit the same behavior.
Error Guessing Technique	Error guessing is a technique in which there is no specific method for identifying the error. It is based on the experience of the test analyst, where the tester uses the experience to guess the problematic areas of the software.
Use Case Technique	Use case Technique used to identify the test cases from the beginning to the end of the system as per the usage of the system. By using this technique, the test team creates a test scenario that can exercise the entire software based on the functionality of each function from start to end.

Time Scheduler Black box testing:



Projects / Time Scheduler / TS-5 / TS-22
1
Attach
Add a child issue
Link issue
To Do
Details
Configure
Activity
Show: All Comments History
Newest first
TP
Add a comment...
Pro tip: press M to comment

Editing mode in Software allows not permissible to edit.

Description

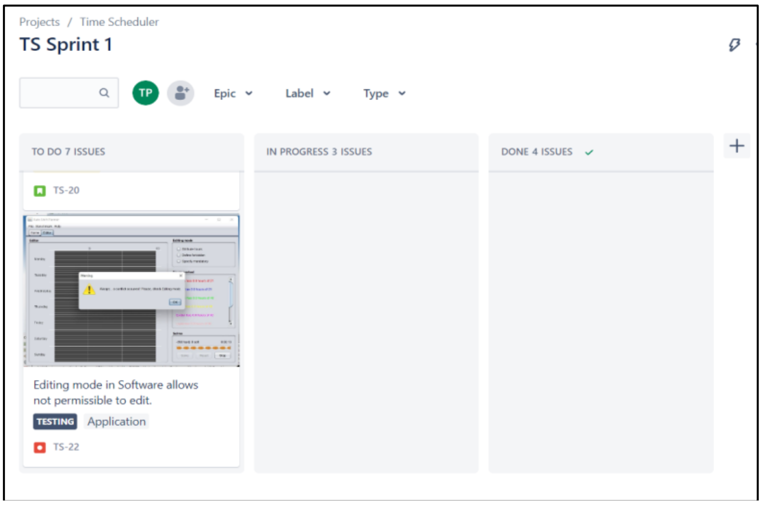
The software consist of editing part which has some permissions linked to it. But the unauthorize are also able to access those features.

Attachments (1)

Screenshot 2022... 29.jfif
27 Sep 2022, 06:04 AM

Assignee	Unassigned Assign to me
Labels	Application
Sprint	TS Sprint 1
Story point estimate	None
Reporter	Tejas Paradkar

Created 13 seconds ago
Updated 13 seconds ago



TS Sprint 1

TO DO 7 ISSUES

IN PROGRESS 3 ISSUES

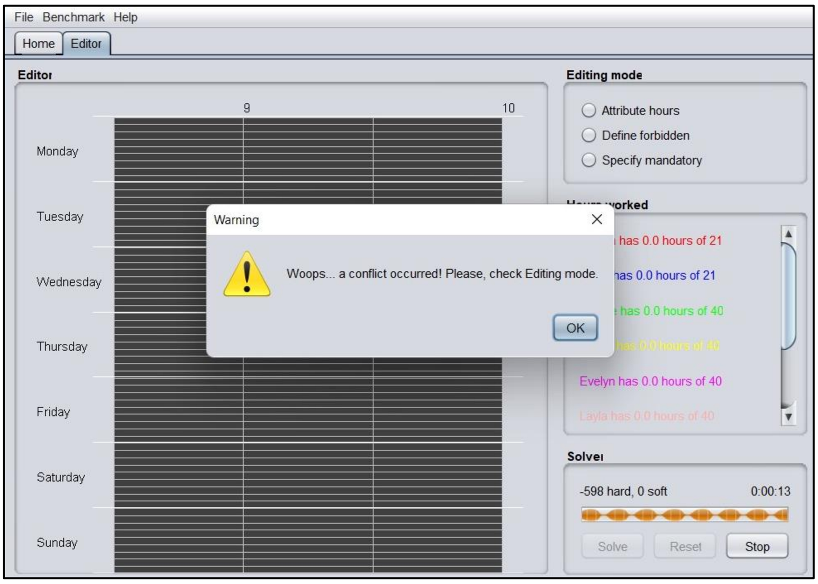
DONE 4 ISSUES

TS-20

Editing mode in Software allows not permissible to edit.

TESTING Application

TS-22



○ Attribute hours
○ Define forbidden
○ Specify mandatory

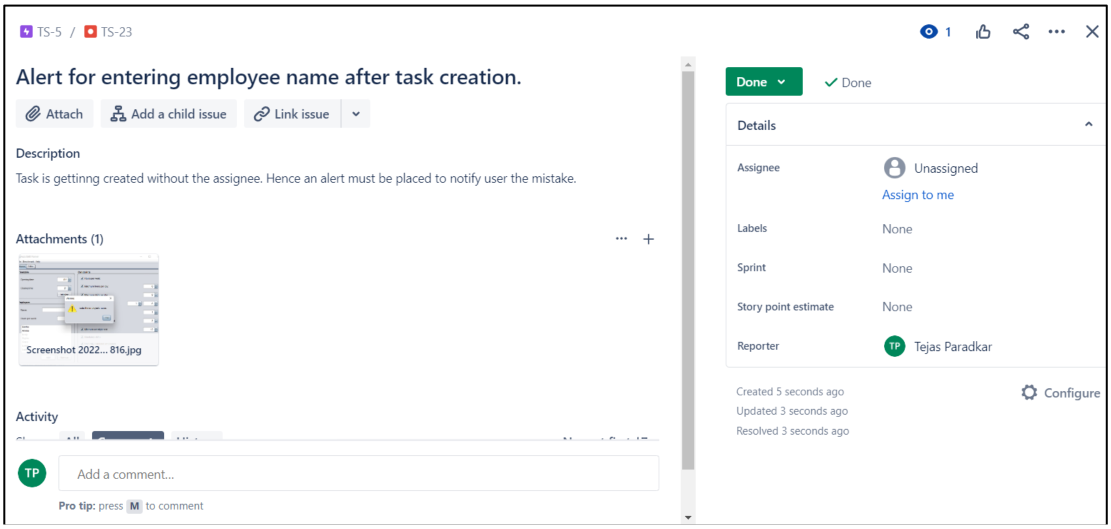
has 0.0 hours of 21
has 0.0 hours of 21
has 0.0 hours of 40
has 0.0 hours of 40
Evelyn has 0.0 hours of 40
Layla has 0.0 hours of 40

-598 hard, 0 soft 0:00:13

Warning

Woops... a conflict occurred! Please, check Editing mode.

OK



TS-5 / TS-23

Alert for entering employee name after task creation.

Description

Task is gettinng created without the assignee. Hence an alert must be placed to notify user the mistake.

Attachments (1)

Screenshot 2022... 816.jpg

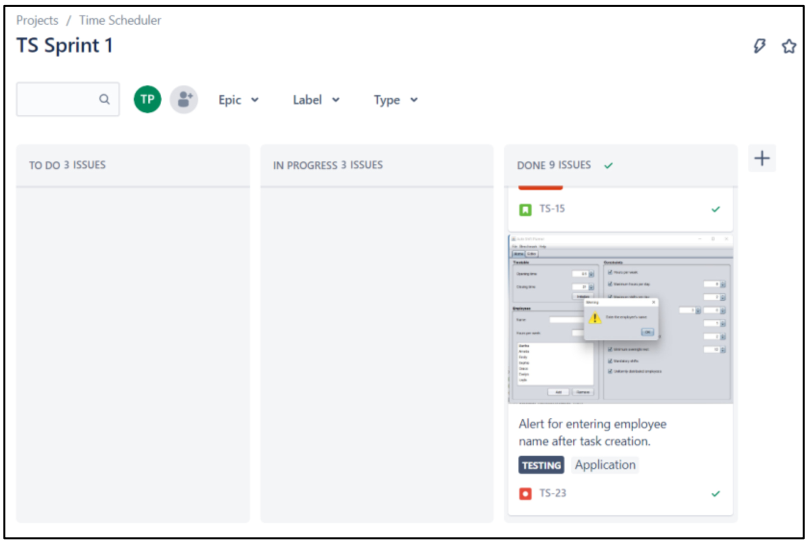
Activity

Add a comment...

Pro tip: press M to comment

Assignee	Unassigned Assign to me
Labels	None
Sprint	None
Story point estimate	None
Reporter	Tejas Paradkar

Created 5 seconds ago
Updated 3 seconds ago
Resolved 3 seconds ago



Projects / Time Scheduler

TS Sprint 1

TO DO 3 ISSUES

IN PROGRESS 3 ISSUES

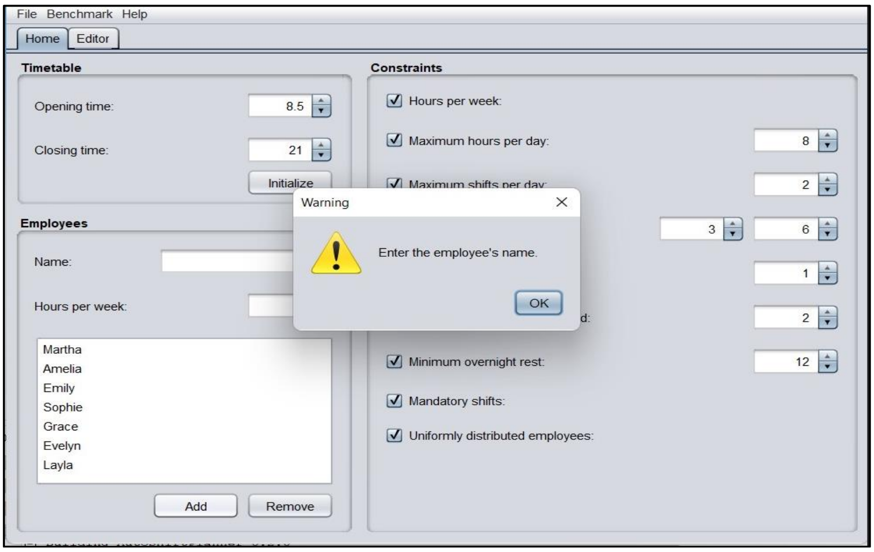
DONE 9 ISSUES ✓

TS-15 ✓

Alert for entering employee name after task creation.

TESTING Application

TS-23 ✓



Timetable

Opening time: 8.5

Closing time: 21

Employees

Name:

Hours per week:

Martha
Amelia
Emily
Sophie
Grace
Evelyn
Layla

Constraints

☑ Hours per week:

☑ Maximum hours per day: 8

☑ Maximum shifts per day: 2

3 6

1

2

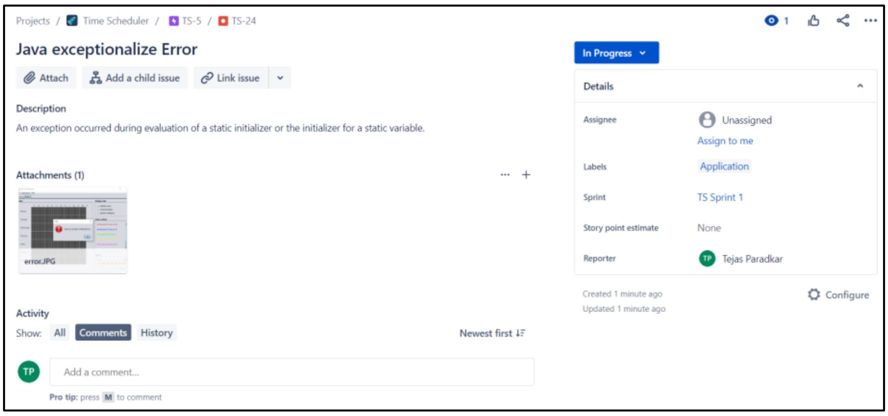
☑ Minimum overnight rest: 12

☑ Mandatory shifts:

☑ Uniformly distributed employees:

Warning

Enter the employee's name.



Projects / Time Scheduler / TS-5 / TS-24

Java exceptionalize Error

Description

An exception occurred during evaluation of a static initializer or the initializer for a static variable.

Attachments (1)

error.JPG

Details

Assignee	Unassigned Assign to me
Labels	Application
Sprint	TS Sprint 1
Story point estimate	None
Reporter	Tejas Paradkar

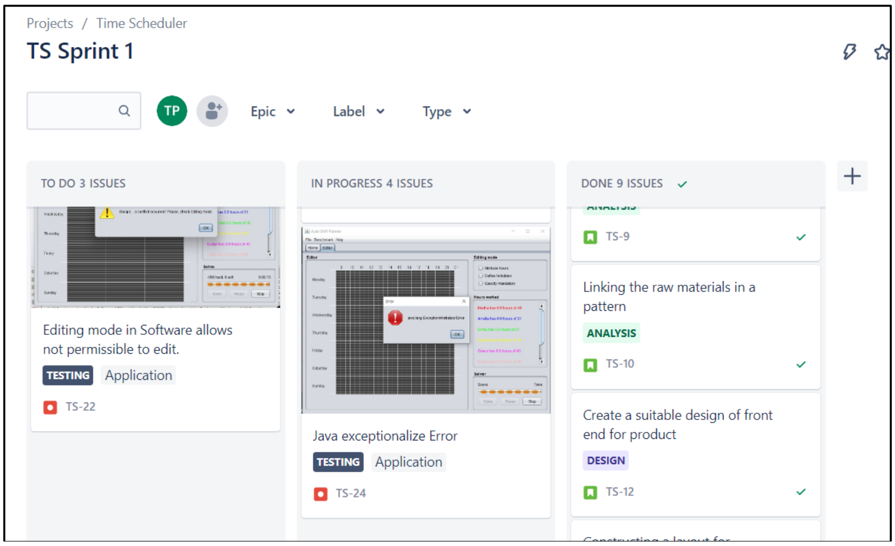
Created 1 minute ago
Updated 1 minute ago

Activity

Show: All Comments History Newest first

Add a comment...

Pro tip: press M to comment



Projects / Time Scheduler

TS Sprint 1

TO DO 3 ISSUES

Editing mode in Software allows not permissible to edit.
TESTING Application
TS-22

IN PROGRESS 4 ISSUES

Java exceptionalize Error
TESTING Application
TS-24

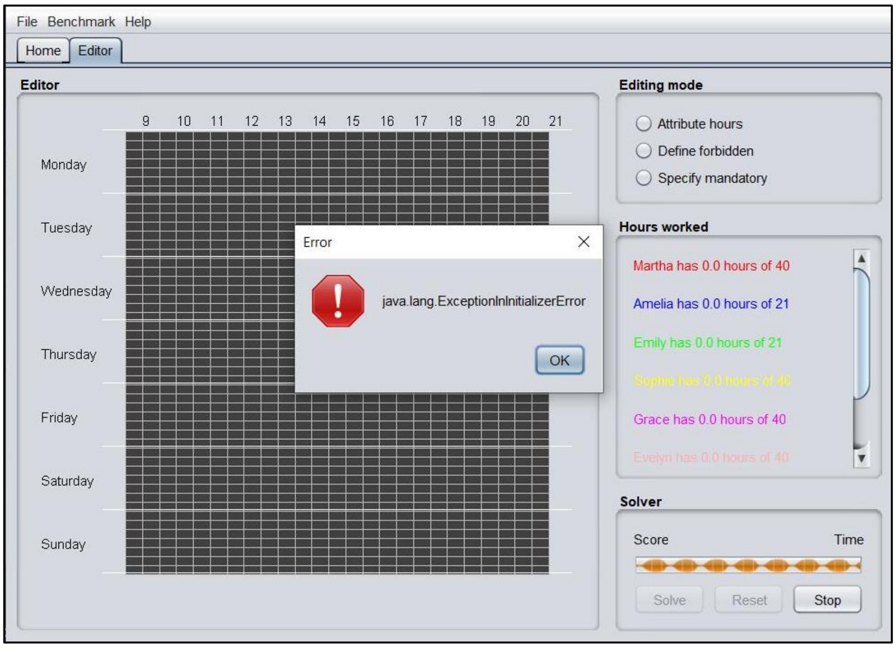
DONE 9 ISSUES ✓

ANALYSIS
TS-9 ✓

Linking the raw materials in a pattern
ANALYSIS
TS-10 ✓

Create a suitable design of front end for product
DESIGN
TS-12 ✓

Constructing a layout for



Error

java.lang.ExceptionInInitializerError

Editing mode

○ Attribute hours
○ Define forbidden
○ Specify mandatory

Hours worked

Martha has 0.0 hours of 40
Amelia has 0.0 hours of 21
Emily has 0.0 hours of 21
Sophie has 0.0 hours of 40
Grace has 0.0 hours of 40
Evelyn has 0.0 hours of 40

Solver

Conclusion: Thus, wrote test cases for black box testing.

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

Experiment No: 8

Aim: Write test cases for white box testing

Software Used: **Selenium/GitHub/Jira**

Theory: -

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

White box testing which also known as glass box is **testing, structural testing, clear box testing, open box testing and transparent box testing**. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

Here, the test engineers will not include in fixing the defects for the following reasons:

- Fixing the bug might interrupt the other features. Therefore, the test engineer should always find the bugs, and developers should still be doing the bug fixes.
- If the test engineers spend most of the time fixing the defects, then they may be unable to find the other bugs in the application.

Techniques Used in White Box Testing

Data Flow Testing	Data flow testing is a group of testing strategies that examines the control flow of programs in order to explore the sequence of variables according to the sequence of events.
Control Flow Testing	Control flow testing determines the execution order of statements or instructions of the program through a control structure. The control structure of a program is used to develop a test case for the program. In this technique, a particular part of a large program is selected by the tester to set the testing path. Test cases represented by the control graph of the program.
Branch Testing	Branch coverage technique is used to cover all branches of the control flow graph. It covers all the possible outcomes (true and false) of each condition of decision point at least once.
Statement Testing	Statement coverage technique is used to design white box test cases. This technique involves execution of all statements of the source code at least once. It is used to calculate the total number of executed statements in the source code, out of total statements present in the source code.
Decision Testing	This technique reports true and false outcomes of Boolean expressions. Whenever there is a possibility of two or more outcomes from the statements like do while statement, if statement and case statement (Control flow statements), it is considered as decision point because there are two outcomes either true or false.

Let us consider the following code:

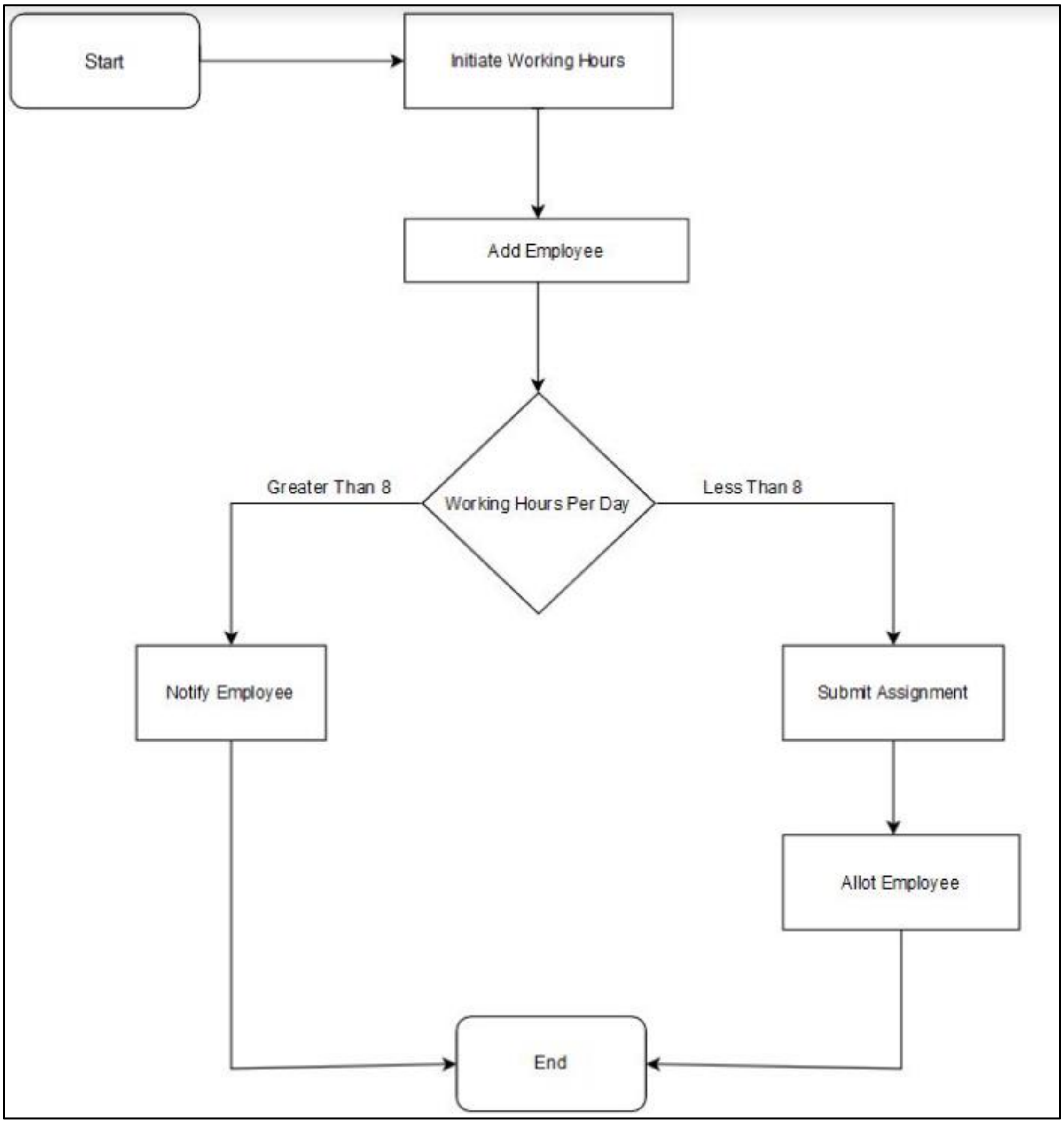
```
INPUT A & B

C = A + B

IF C>100

PRINT “ITS DONE”
```

Now in the first, line, we assign the value of A and B. Let us suppose A = 60 and B = 50. Moving on to the second line, now C is assigned a value of A+B, here A = 60 and B = 50, hence C = 110. Moving on to the third line, we will check if C > 100, here the condition is true and hence we should get our result as ITS DONE

White Box testing on Time Scheduler:

Case 1: Working hours calculation and work allotment for entity.

Flowchart:



Pseudocode:

```
Input working hours,
C = working hours,
If C>8
Print “Work Done”;
Else
Print “New Task”;
```

Case 2: Number of leaves taken

Pseudocode:

```
A = 5 (leaves taken till now)
B = 12 (Total leaves available)
Read A, B
If (A>B)
    X=A+1
    PRINT “Leave granted”
Else
    X = B-1
    PRINT “Exceeded the leave, no leave granted”
```

Conclusion: Thus, wrote test cases for white box testing

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

Experiment No: 9

Aim: Preparation of Risk Mitigation, Monitoring and Management plan (RMMM).

Software Used: MS Word

Theory: -

RMMM Plan:

A risk management technique is usually seen in the software Project plan. This can be divided into Risk Mitigation, Monitoring, and Management Plan (RMMM). In this plan, all works are done as part of risk analysis. As part of the overall project plan project manager generally uses this RMMM plan.
In some software teams, risk is documented with the help of a Risk Information Sheet (RIS). This RIS is controlled by using a database system for easier management of information i.e creation, priority ordering, searching, and other analysis. After documentation of RMMM and start of a project, risk mitigation and monitoring steps will start.

Risk Mitigation:

It is an activity used to avoid problems (Risk Avoidance).
Steps for mitigating the risks as follows.

1. Finding out the risk.
2. Removing causes that are the reason for risk creation.
3. Controlling the corresponding documents from time to time.
4. Conducting timely reviews to speed up the work.

Risk Monitoring:

It is an activity used for project tracking.
It has the following primary objectives as follows.

1. To check if predicted risks occur or not.
2. To ensure proper application of risk aversion steps defined for risk.
3. To collect data for future risk analysis.
4. To allocate what problems are caused by which risks throughout the project.

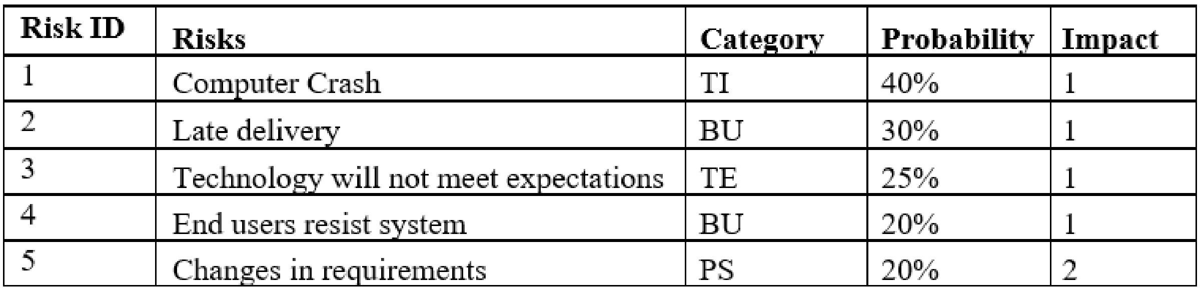
Risk Management and planning:

It assumes that the mitigation activity failed and the risk is a reality. This task is done by Project manager when risk becomes reality and causes severe problems. If the project manager effectively uses project mitigation to remove risks successfully then it is easier to manage the risks. This shows that the response that will be taken for each risk by a manager. The main objective of the risk management plan is the risk register. This risk register describes and focuses on the predicted threats to a software project.

Steps for Risk Management

1. Identify possible risks and recognize what can go wrong
2. Analyze each risk to estimate the probability that it will occur and the impact (i.e., damage) that it will do if it does occur.
3. Rank the risks by probability and impact. Impact may be negligible, marginal, critical, and catastrophic.
4. Develop a contingency plan to manage those risks having high probability and high impact

Risk Table



Risk ID	Risks	Category	Probability	Impact
1	Computer Crash	TI	40%	1
2	Late delivery	BU	30%	1
3	Technology will not meet expectations	TE	25%	1
4	End users resist system	BU	20%	1
5	Changes in requirements	PS	20%	2

Impact Values:

1 – Catastrophic, 2 – Critical, 3 – Marginal, 4 – Negligible

Risk: Computer Crash

- **Mitigation**

The cost associated with a computer crash resulting in a loss of data is crucial. A computer crash itself is not crucial, but rather the loss of data. A loss of data will result in not being able to deliver the product to the customer. This will result in a not receiving a letter of acceptance from the customer. Without the letter of acceptance, the group will receive a failing grade for the course. As a result the organization is taking steps to make multiple backup copies of the software in development and all documentation associated with it, in multiple locations.

- **Monitoring**

When working on the product or documentation, the staff member should always be aware of the stability of the computing environment they're working in. Any changes in the stability of the environment should be recognized and taken seriously.

- **Management**

The lack of a stable-computing environment is extremely hazardous to a software development team. If the computing environment is found unstable, the development team.

Risk: Late Delivery

- **Mitigation**

  The cost associated with a late delivery is critical. A late delivery will result in a late delivery of a letter of acceptance from the customer. Without the letter of acceptance, the group will receive a failing grade for the course. Steps have been taken to ensure a timely delivery by gauging the scope of project based on the delivery deadline.

- **Monitoring**

  A schedule has been established to monitor project status. Falling behind schedule would indicate a potential for late delivery. The schedule will be followed closely during all development stages.

- **Management**

  Late delivery would be a catastrophic failure in the project development. If the project cannot be delivered on time the development team will not pass the course. If it becomes apparent that the project will not be completed on time, the only course of action available would be to request an extension to the deadline form the customer.

Risk: Technology Does Not Meet Specifications

- **Mitigation**

  In order to prevent this from happening, meetings (formal and informal) will be held with the customer on a routine business. This ensures that the product we are producing, and the specifications of the customer are equivalent.

- **Monitoring**

  The meetings with the customer should ensure that the customer and our organization understand each other and the requirements for the product.

- **Management**

  Should the development team come to the realization that their idea of the product specifications differs from those of the customer, the customer should be immediately notified and whatever steps necessary to rectify this problem should be done. Preferably a meeting should be held between the development team and the customer to discuss at length this issue.

Risk: End Users Resist System

- **Mitigation**

  In order to prevent this from happening, the software will be developed with the end user in mind. The user-interface will be designed in a way to make use of the program convenient and pleasurable.

- **Monitoring**

  The software will be developed with the end user in mind. The development team will ask the opinion of various outside sources throughout the development phases. Specifically the user-interface developer will be sure to get a thorough opinion from others.

- **Management**

  Should the program be resisted by the end user, the program will be thoroughly examined to find the reasons that this is so. Specifically the user interface will be investigated and if necessary, revamped into a solution.

Risk: Changes in Requirements

- **Mitigation**

  In order to prevent this from happening, meetings (formal and informal) will be held with the customer on a routine business. This ensures that the product we are producing, and the requirements of the customer are equivalent.

- **Monitoring**

  The meetings with the customer should ensure that the customer and our organization understand each other and the requirements for the product.

- **Management**

  Should the development team come to the realization that their idea of the product requirements differs from those of the customer, the customer should be immediately notified and whatever steps necessary to rectify this problem should be taken. Preferably a meeting should be held between the development team and the customer to discuss at length this issue.

Conclusion: Thus, prepared Risk Mitigation, Monitoring and Management plan (RMMM.)

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				

Experiment No: 10

Aim: Version control of the project

Software Used: GitHub

Theory: -

Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done to the code.

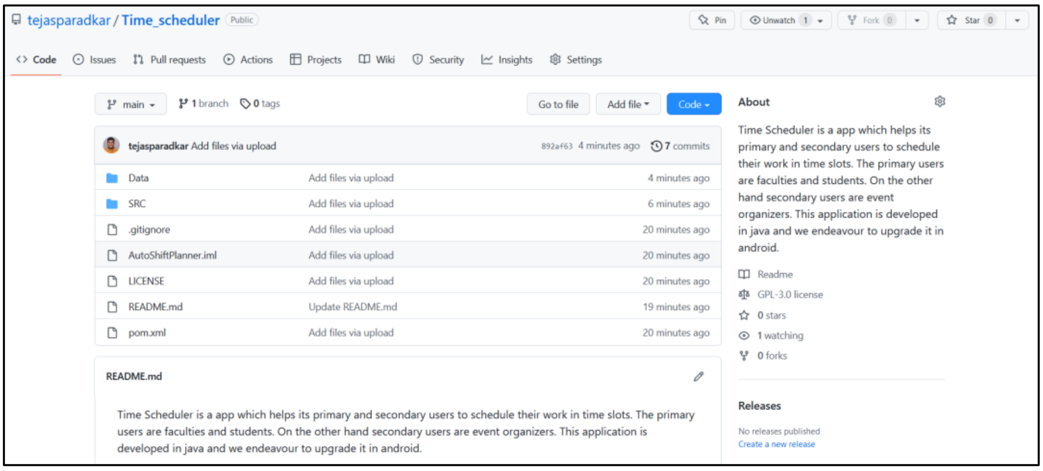
As we know that a software product is developed in collaboration by a group of developers they might be located at different locations and each one of them contributes in some specific kind of functionality/features. So in order to contribute to the product, they made modifications in the source code(either by adding or removing). A version control system is a kind of software that helps the developer team to efficiently communicate and manage (track) all the changes that have been made to the source code along with the information like who made and what change has been made. A separate branch is created for every contributor who made the changes and the changes aren't merged into the original source code unless all are analyzed as soon as the changes are green signaled they merged to the main source code. It not only keeps source code organized but also improves productivity by making the development process smooth.

Benefits of the version control system:
a) Enhances the project development speed by providing efficient collaboration,
b) Leverages the productivity, expedite product delivery, and skills of the employees through better communication and assistance,
c) Reduce possibilities of errors and conflicts meanwhile project development through traceability to every small change,
d) Employees or contributor of the project can contribute from anywhere irrespective of the different geographical locations through this **VCS,**
e) For each different contributor of the project a different working copy is maintained and not merged to the main file unless the working copy is validated. A most popular example is **Git, Helix core, Microsoft TFS,**
f) Helps in recovery in case of any disaster or contingent situation,
g) Informs us about Who, What, When, Why changes have been made.

Use of Version Control System:

- **A repository:** It can be thought of as a database of changes. It contains all the edits and historical versions (snapshots) of the project.
- **Copy of Work (sometimes called as checkout):** It is the personal copy of all the files in a project. You can edit to this copy, without affecting the work of others and you can finally commit your changes to a repository when you are done making your changes.

Our Mini Project (Time Scheduler) is uploaded on GitHub as a repository for version control.



tejasparadkar Add files via upload — 892af63 4 minutes ago — 7 commits

Data	Add files via upload	4 minutes ago
SRC	Add files via upload	6 minutes ago
.gitignore	Add files via upload	20 minutes ago
AutoShiftPlanner.iml	Add files via upload	20 minutes ago
LICENSE	Add files via upload	20 minutes ago
README.md	Update README.md	19 minutes ago
pom.xml	Add files via upload	20 minutes ago

README.md

Time Scheduler is a app which helps its primary and secondary users to schedule their work in time slots. The primary users are faculties and students. On the other hand secondary users are event organizers. This application is developed in java and we endeavour to upgrade it in android.

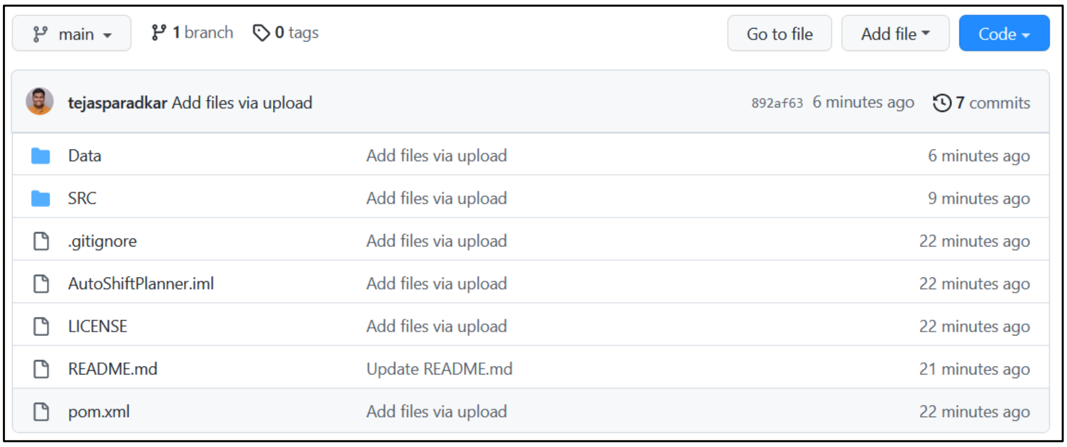
About

Time Scheduler is a app which helps its primary and secondary users to schedule their work in time slots. The primary users are faculties and students. On the other hand secondary users are event organizers. This application is developed in java and we endeavour to upgrade it in android.

Readme
GPL-3.0 license
0 stars
1 watching
0 forks

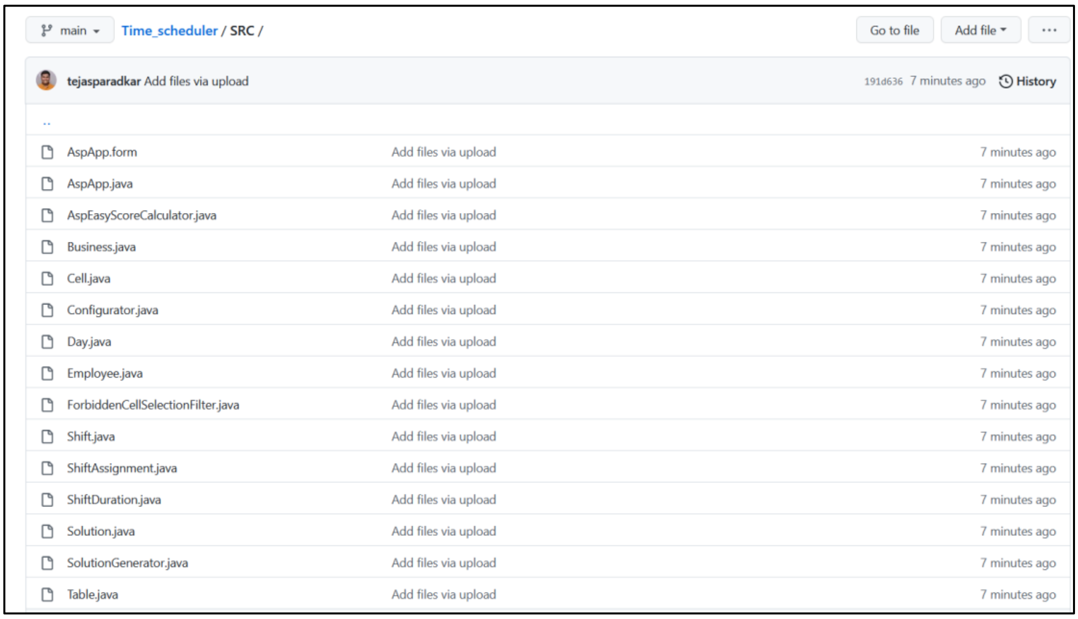
Releases

No releases published
Create a new release



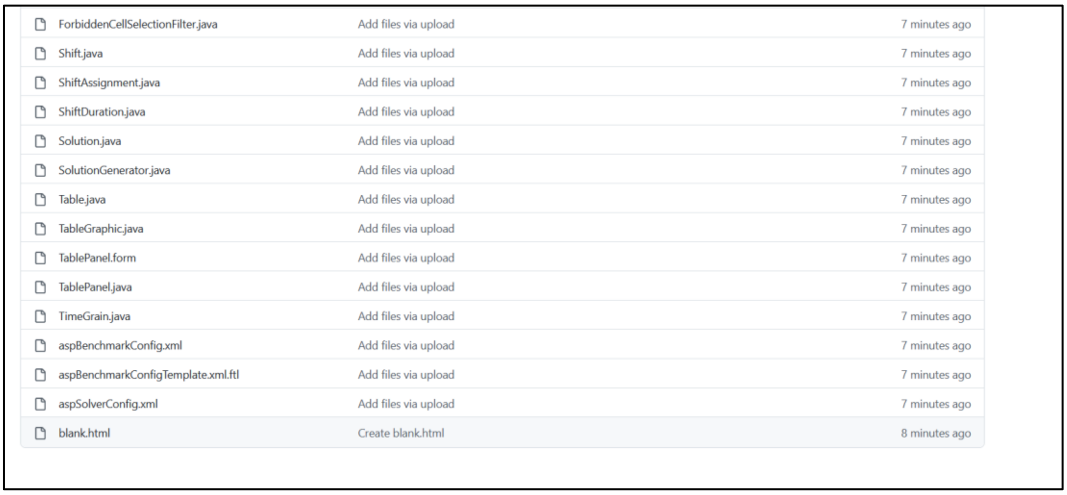
tejasparadkar Add files via upload — 892af63 6 minutes ago — 7 commits

Data	Add files via upload	6 minutes ago
SRC	Add files via upload	9 minutes ago
.gitignore	Add files via upload	22 minutes ago
AutoShiftPlanner.iml	Add files via upload	22 minutes ago
LICENSE	Add files via upload	22 minutes ago
README.md	Update README.md	21 minutes ago
pom.xml	Add files via upload	22 minutes ago

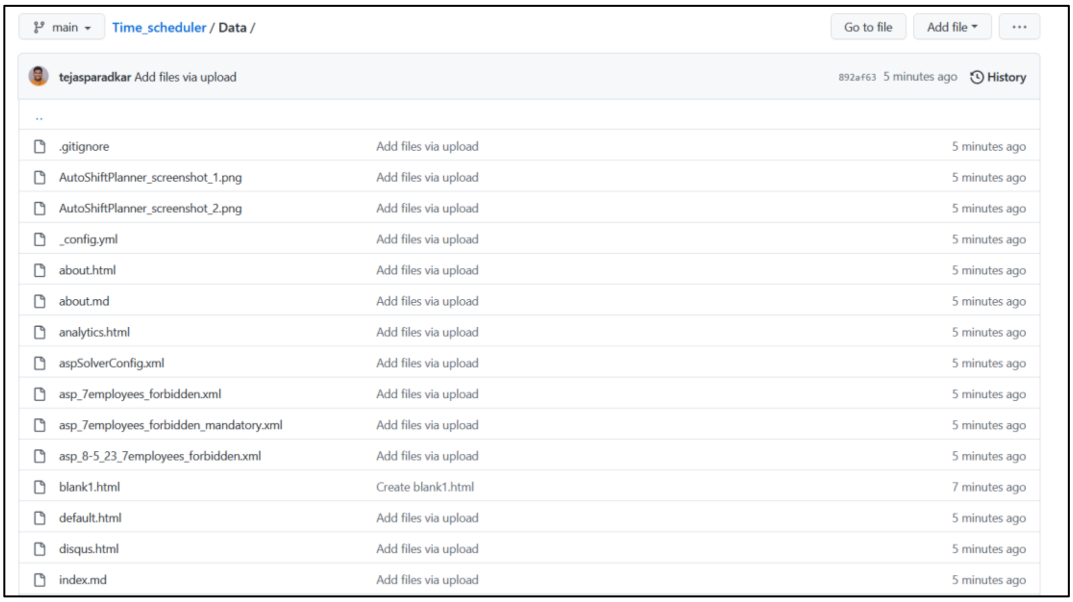


tejasparadkar Add files via upload — 191d636 7 minutes ago — History

..		
AspApp.form	Add files via upload	7 minutes ago
AspApp.java	Add files via upload	7 minutes ago
AspEasyScoreCalculator.java	Add files via upload	7 minutes ago
Business.java	Add files via upload	7 minutes ago
Cell.java	Add files via upload	7 minutes ago
Configurator.java	Add files via upload	7 minutes ago
Day.java	Add files via upload	7 minutes ago
Employee.java	Add files via upload	7 minutes ago
ForbiddenCellSelectionFilter.java	Add files via upload	7 minutes ago
Shift.java	Add files via upload	7 minutes ago
ShiftAssignment.java	Add files via upload	7 minutes ago
ShiftDuration.java	Add files via upload	7 minutes ago
Solution.java	Add files via upload	7 minutes ago
SolutionGenerator.java	Add files via upload	7 minutes ago
Table.java	Add files via upload	7 minutes ago



ForbiddenCellSelectionFilter.java	Add files via upload	7 minutes ago
Shift.java	Add files via upload	7 minutes ago
ShiftAssignment.java	Add files via upload	7 minutes ago
ShiftDuration.java	Add files via upload	7 minutes ago
Solution.java	Add files via upload	7 minutes ago
SolutionGenerator.java	Add files via upload	7 minutes ago
Table.java	Add files via upload	7 minutes ago
TableGraphic.java	Add files via upload	7 minutes ago
TablePanel.form	Add files via upload	7 minutes ago
TablePanel.java	Add files via upload	7 minutes ago
TimeGrain.java	Add files via upload	7 minutes ago
aspBenchmarkConfig.xml	Add files via upload	7 minutes ago
aspBenchmarkConfigTemplate.xml.ftl	Add files via upload	7 minutes ago
aspSolverConfig.xml	Add files via upload	7 minutes ago
blank.html	Create blank.html	8 minutes ago



Time_scheduler / Data /

tejasparadkar Add files via upload 892af63 5 minutes ago

..

.gitignore	Add files via upload	5 minutes ago
AutoShiftPlanner_screenshot_1.png	Add files via upload	5 minutes ago
AutoShiftPlanner_screenshot_2.png	Add files via upload	5 minutes ago
_config.yml	Add files via upload	5 minutes ago
about.html	Add files via upload	5 minutes ago
about.md	Add files via upload	5 minutes ago
analytics.html	Add files via upload	5 minutes ago
aspSolverConfig.xml	Add files via upload	5 minutes ago
asp_7employees_forbidden.xml	Add files via upload	5 minutes ago
asp_7employees_forbidden_mandatory.xml	Add files via upload	5 minutes ago
asp_8-5_23_7employees_forbidden.xml	Add files via upload	5 minutes ago
blank1.html	Create blank1.html	7 minutes ago
default.html	Add files via upload	5 minutes ago
disqus.html	Add files via upload	5 minutes ago
index.md	Add files via upload	5 minutes ago

Conclusion: Thus performed, version control of the project.

Rubrics →	**Timely Submission (5)**	**Neatness (5)**	**Knowledge (5)**	**Signature of staff With Date**
Marks →				