



insecTo_destrucTo

14.09.2022

Team X-4B

Aanchal Agarwal (PES2UG20CS005)

Ananya Adiga (PES2UG20CS043)

Anirudh Chakravathy Kumar (PES2UG20CS049)

Adarsh Liju Abraham (PES2UG20CS017)

1. Incremental Model

We are using the incremental model to implement the bug tracking software. This model is a **SE model** in which the requirements are identified in the first phase itself and each requirement is **divided into modules**. Each module goes through the requirements, design, implementation and testing phases and each module is added to the previous release during **subsequent releases**.

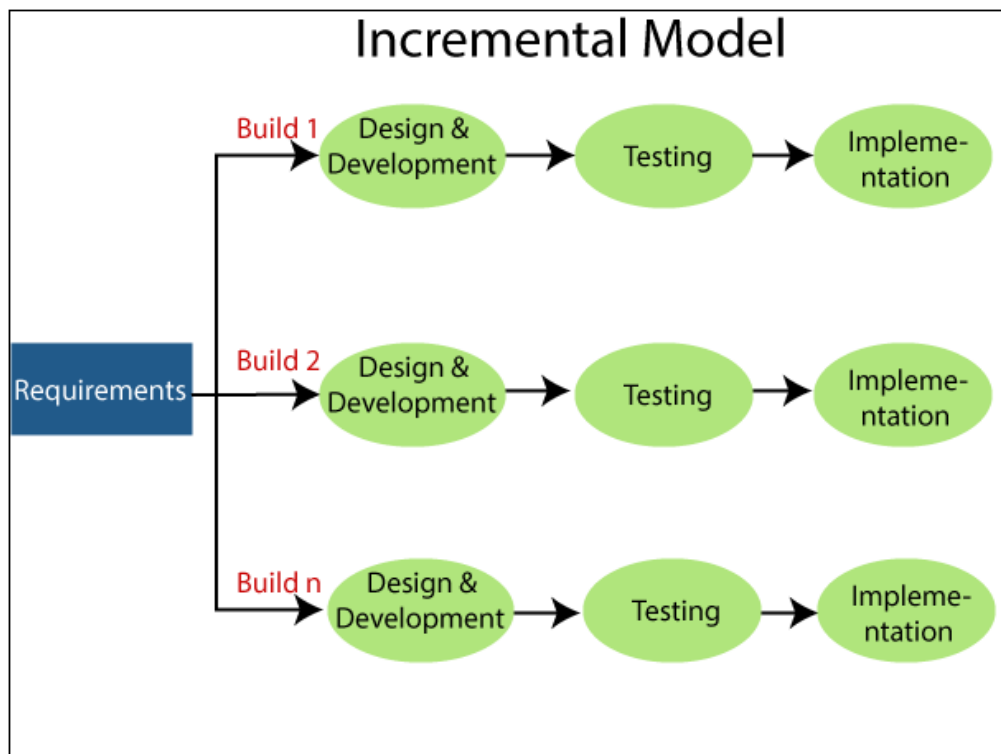
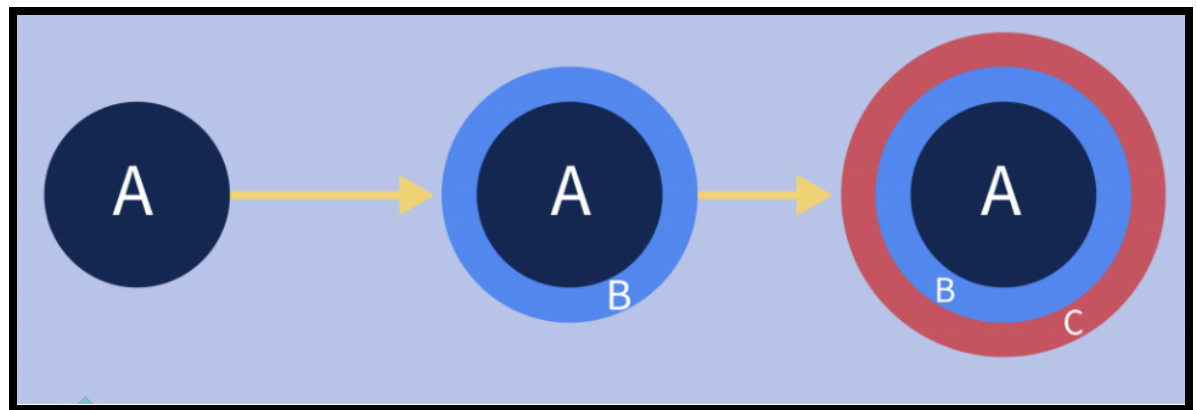


Fig- 1.1 - Incremental model cycle

- The main reason to choose incremental model is that it is **flexible** i.e. the requirements are not frozen unlike the waterfall model
- Early versions are delivered and each increment acts as a **prototype**
- These prototypes help **detect errors** at an early stage and fix them to avoid loss due to a defective software and provides better **risk mitigation**
- Good planning and prioritized iterations ensure reduction of **over functionality**
- Since our team has got the **bare bone and skeletal structure** of the project in hand during the lifecycle phase, **prioritizing our functionality** and adding modules in each subsequent release increases team **productivity**.
- Since we have **clear ideas** about the deliverables it allows us to break down the requirements to **modules**.

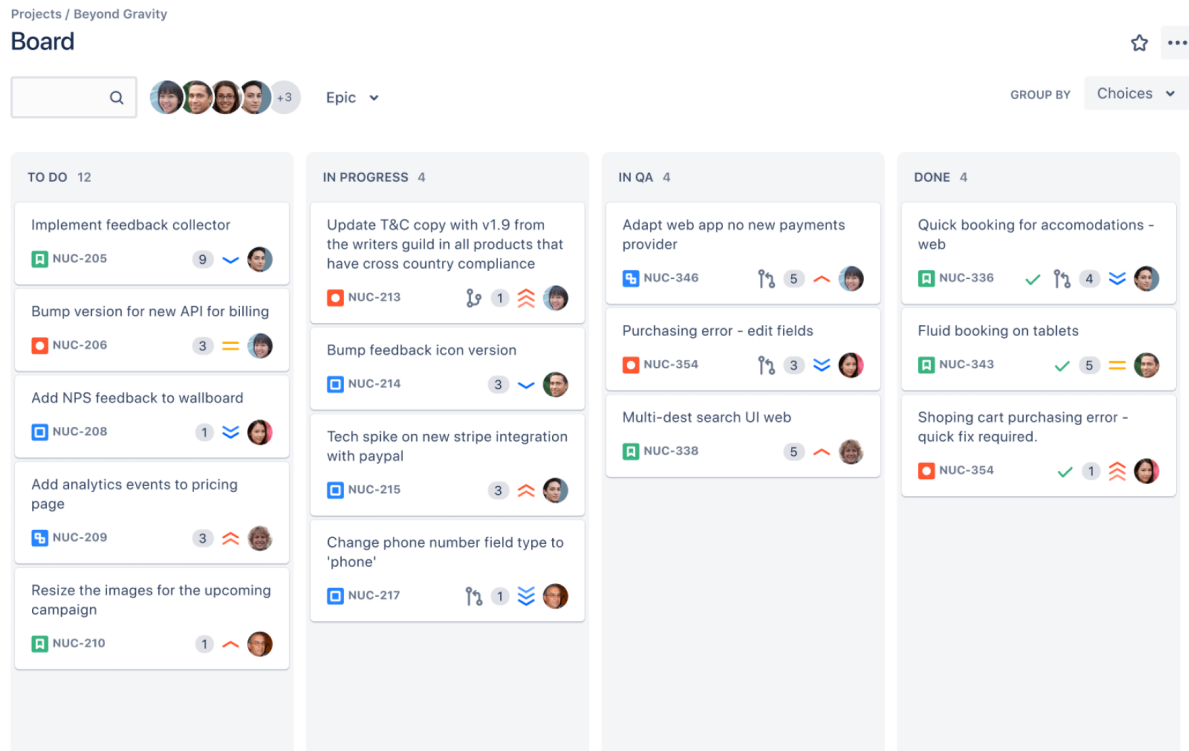
- **Testing and debugging is easier** and feasible for small iterations which ensures that all our requirements are met under the available time constraint
- The **client can review** each release to check if their requirements are met and request for more functionalities if any which makes it easier for the development team to implement them as another iteration as a **backlog** rather than implementing after the product is ready for deployment, this ensures **customer satisfaction**.
- And lastly the incremental model is more **cost effective** compared to the other models.



Planning Tools



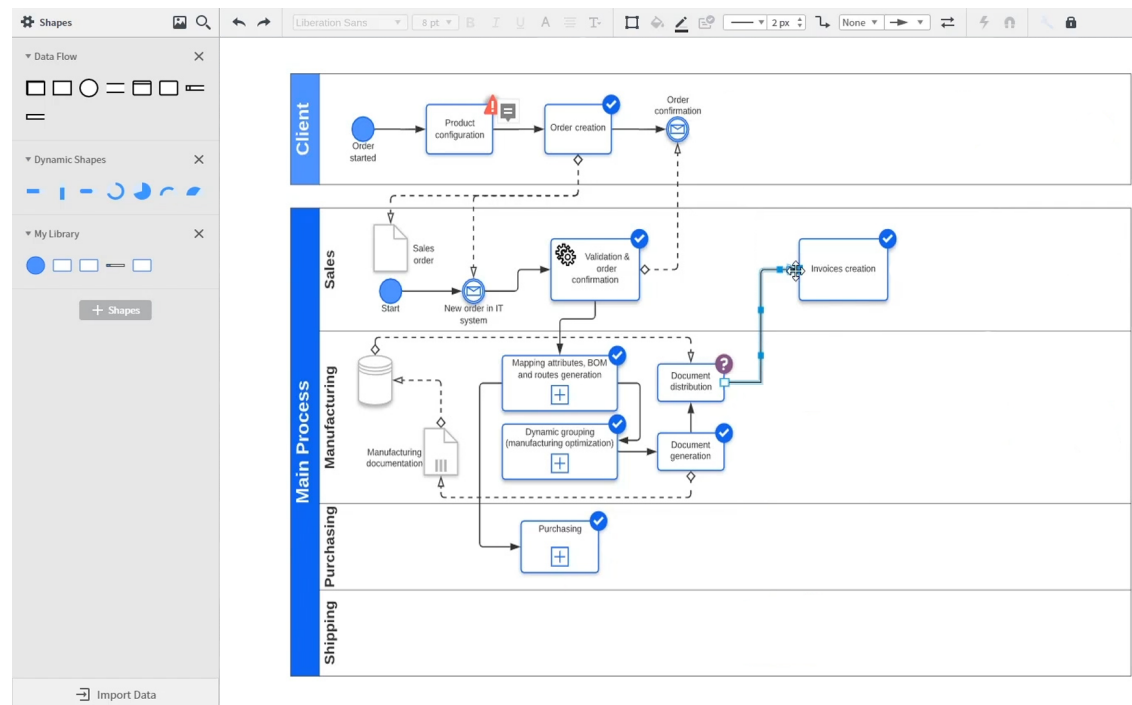
- Jira is a proprietary issue tracking product developed by **Atlassian** that allows **bug tracking** and **agile project management**.
- Jira can be configured to fit any type of project. Teams can start with a project template or create their **own custom workflow**.
- Jira issues, also known as **tasks**, track each piece of work that needs to pass through the workflow steps to completion. Customizable permissions enable admins to determine who can see and perform which actions.
- With all project information in place, **reports** can be generated to track progress, productivity, and ensure nothing slips.
- Jira provides **planning and roadmap tools** so teams can manage stakeholders, budgets, and feature requirements from day one.
- Jira integrates with a variety of **CI/CD tools** to facilitate **transparency** throughout the software development life cycle.
- When it's ready to deploy, **live production code status** information is surfaced in the Jira issue. Integrated feature flagging tools allow teams to roll out new features gradually and safely.
- A **backlog** is a set of activities or issues that the team needs to be resolved within a specific iteration. All the issues of your project are grouped in a **backlog and sprint**.
- In scrum backlog, you can perform multiple activities such as create and update issues, drag and drop the issues to prioritize them, assign them to sprints, epics, versions, manage epics, etc.
- **Backlogs** can be easily created in Jira Software which is integrated into the system by default.



Design Tool



- **Lucidchart** is a **web-based diagramming** application that allows users to visually collaborate on **drawing, revising** and **sharing charts** and diagrams, and improve **processes, systems, and organizational** structures.
- The following are the reasons we choose **LucidChart** :-
 - A drag-and-drop interface
 - Real-time co-authoring, shape-specific comments, and collaborative cursors
 - Data linking
 - Auto-visualization to generate org charts and ERDs
 - SQL import and export capabilities
- LucidChart also integrates well with **Jira Software** which is a part of our main toolchain
- The simple layout, easy sharing, and the ability to build complex flowcharts make Lucidchart the best out there.

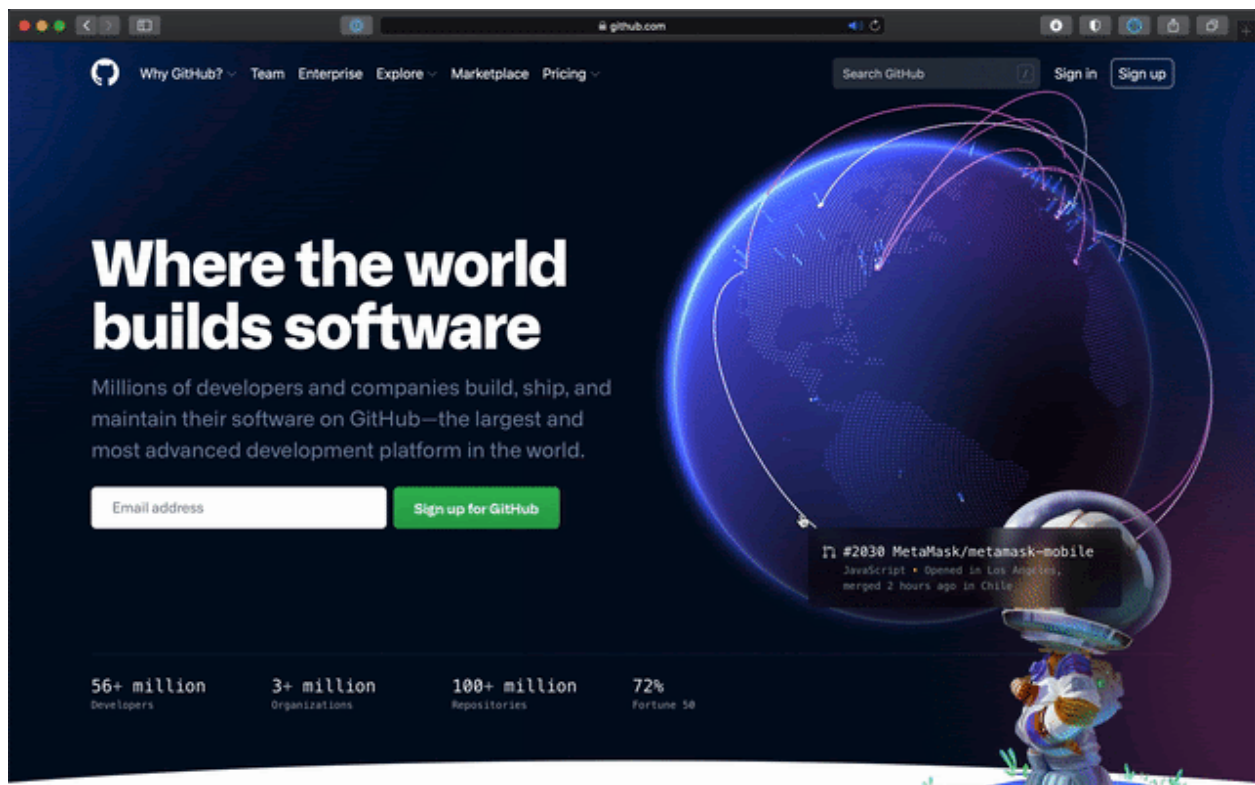


Version Control



- **GitHub, Inc.**, is an Internet hosting service for software development and version control using **Git**.
- It provides the distributed version control of Git plus **access control**, **bug tracking**, software feature requests, task management, **continuous integration**, and **wikis** for every project.
- It is commonly used to host open source software development projects. It is used by **noobs** and **power users** together.
- GitHub supports the following formats and features:
 - Documentation, including automatically rendered README files in a variety of Markdown-like file formats
 - Wikis
 - GitHub Actions, which allows building continuous integration and continuous deployment pipelines for testing, releasing and deploying software without the use of third-party websites/platforms

- Graphs: pulse, contributors, commits, code frequency, punch card, network, members
- Integrations Directory
- Email notifications
- Discussions
- Option to subscribe someone to notifications by @ mentioning them.
- Emojis
- Nested task-lists within files
- Visualization of geospatial data
- 3D render files that can be previewed using a new integrated STL file viewer that displays the files on a "3D canvas." [75] The viewer is powered by WebGL and Three.js.
- Photoshop's native PSD format can be previewed and compared to previous versions of the same file.
- PDF document viewer
- Security Alerts of known Common Vulnerabilities and Exposures in different packages



Bug Tracking



Mantis Bug Tracker

- MantisBT makes **collaboration** with team members & clients **easy, fast, and professional**.
- MantisBT is an **open source issue tracker** that provides a delicate balance between simplicity and power. Users are able to get started in minutes and start **managing** their projects while collaborating with their **teammates and clients** effectively.
- The most common use of MantisBT is to track **software defects**. However, MantisBT is often configured by users to serve as a **more generic issue tracking system** and **project management tool**.



Anonymous | [Login](#) | [Signup for a new account](#)

2009-02-01 03:44 EST



[Main](#) | [My View](#) | [View Issues](#) | [Change Log](#) | [Roadmap](#) | [Docs](#) | [Wiki](#) | [ManTweet](#) | [Repositories](#)

[Jump](#)

Unassigned [^] (1 - 10 / 2006)

0010092	Logging of user action (login, download, ...) feature - 2009-01-30 09:55
0009394	MantisConnect Webservice crashes when trying to get issue information api soap - 2009-01-30 07:43
0010094	EEmail generation if not assigned bugtracker - 2009-01-30 07:02
0010091	CHANGE STATUS TO: Assigned vs ASSIGN TO ?? customization - 2009-01-29 23:37
0004640	New custom field types: "Version", "User" custom fields - 2009-01-29 22:52
0003790	Additional Custom-Field-Type "users" custom fields - 2009-01-29 22:49
0010065	Custom field "date" not saved silently custom fields - 2009-01-29 08:22
0006418	Change Resolve status and Assignment at the same time feature - 2009-01-29 05:41
0007104	Auto assign on resolve bugtracker - 2009-01-29 05:39
0007150	Automatic reassignment on status transition feature - 2009-01-29 05:39

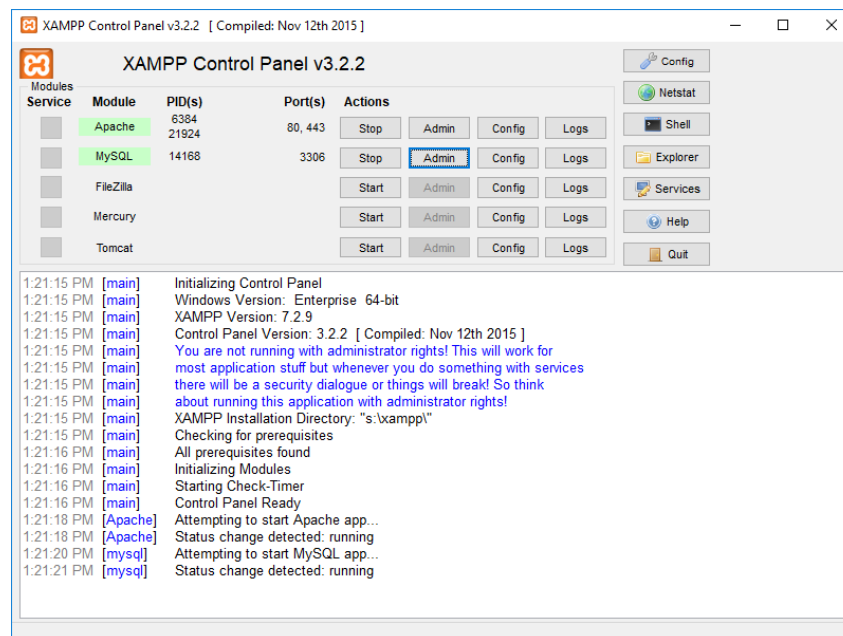
Resolved [^] (1 - 10 / 254)

0010093	(phpmailer) class.phpmailer.php => private \$smtp is accessed by mantis email - 2009-01-31 14:37
0010056	Category national character upgrade - 2009-01-30 14:30
0009455	Database configuration does not correctly deal with array administration - 2009-01-30 11:18
0010089	Updating version field does not trigger history note administration - 2009-01-28 03:55
0008638	The "Show Content" link on attached tx file doesn't show text webpage - 2009-01-28 03:47
0007790	Links protected by brackets are not processed properly bugtracker - 2009-01-27 12:27
0009216	Problems with Russian language localization - 2009-01-27 11:40
0007144	Unable to set realname because of existing username administration - 2009-01-25 14:21
0010078	logging into the site with out entering login details security - 2009-01-24 12:44
0010073	Form Validate in JavaScript javascript - 2009-01-22 14:43

Development and Testing Tool



- XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends Consisting mainly of **the Apache HTTP Server, MariaDB database**, and interpreters for scripts written in the PHP and Perl programming languages.
- Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.
- XAMPP is regularly updated to the latest releases of **Apache, MariaDB, PHP and Perl**. It also comes with a number of other modules, including **OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress** and more.
- **Self-contained, multiple instances** of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a **full and a standard version** (Smaller version).



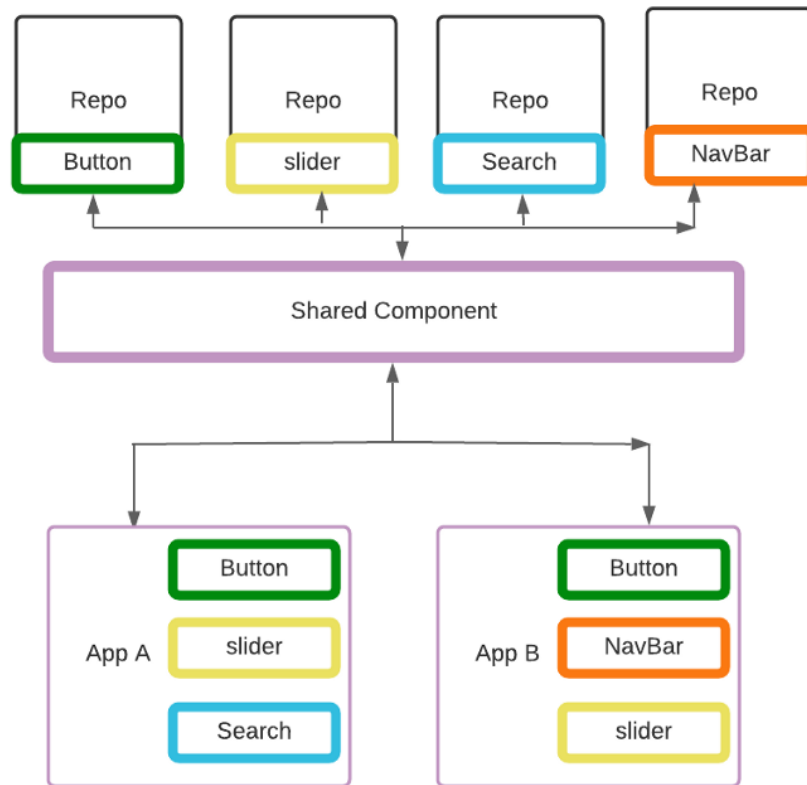
Deliverables



- A deliverable is a tangible or intangible good or service produced as a result of a project that is intended to be delivered to a customer (either internal or external).
- A deliverable may be composed of multiple smaller deliverables. It may be either an outcome to be achieved (as in "The corporation says that becoming profitable this year is a deliverable") or an output to be provided (as in "The deliverable for the completed project consists of a special-purpose electronic device and its controlling software").

A Deep Dive into Reusable Components.

- Reusability is one of the most common and frequently used buzzwords in software development.
- Developers are building components based on collections of reusable components, and recently most of the technologies started to equip themselves with this function.
- In computer science and software engineering, reusability is the use of existing assets in some form within the software product development process; these assets are products and by-products of the software development life cycle and include code, software components, test suites, designs and documentation.



How Code is reused in our project

The above image depicts the importance of reusability with the concept of shared reusable components and how efficiently it can be used to build, distribute, and collaborate over components to build multiple applications and projects.

Why do we use reusable components ?

- Time, Cost and Effort.
- Create a new higher-level abstraction to development.
- Multiple Formats are trending everywhere and need to balance out this.
- Better Usability, there is always demand for simpler and easier UI.
- Reduce redundancy, consistent brand expectation and maximize developer productivity.
- Easier testing.

Components to Reuse	Components to Build
Cross platform	Manage accounts
Robust and real time	Store details in database
Less response time	High throughput
Security	Reliability
One Bug , One Member rule	Update Issue status
Ticket number for each issue	Notifications for user

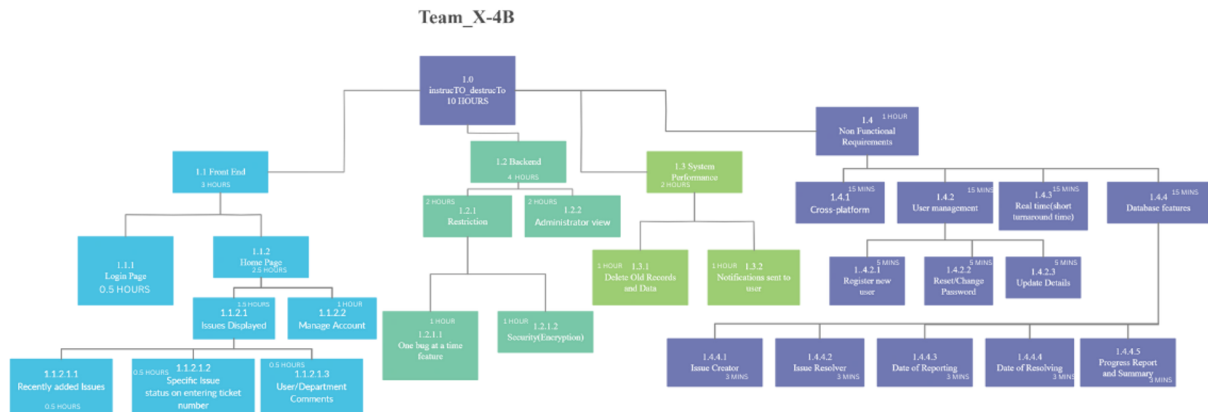
FUNCTIONAL DELIVERABLES :

- **Security-** most of the softwares available in industry provide high-level security to user details , this existing feature can be reused so that it increases team productivity
- **Restrict** the bug to be resolved by only one member- The bare bone of this feature is in our hand which can be implemented in any software such as to restrict user access
- **Store** the most common errors and its solution- A database needs to be built to store error information
- **Delete old records** and data - The records of the database built needs to be deleted and updated frequently to store data efficiently
- Automatic **notifications** to the user- the notifications declaring issue status to both user and team needs to be built
- **Update issue status** - Each time there is a progress in bug fixing the status should be modified and notified that needs to be built
- **Generating ticket number** for each issue- generating ticket number is nothing but generating a unique number each time and that piece of code is already in hand and can be reused.

NON- FUNCTIONAL DELIVERABLES-

- **Cross platform** - If this app can be made cross platform by any existing framework users can follow up from any device . This indirectly increases productivity as many people around the world now have access to more devices.
- **Robust and real time** - It should be strong from inside and outside , fast and concurrent with current time zone , this can help in dealing with foriegn clients who live in a different timezone
- **Manage accounts** - An administrator should be given full access to all accounts , creating, deleting and updating issues and users. Users should have some restrictions depending on the constraints.
- **Store user and issue details in a database** - An SQL database is used because it's good in deployment and scalability. DBMS which are SQL driven are global and stable on runtime.
- **Less response time** - The website and backend should be insanely fast as users time is very important and since our project is agile based , we value customers
- **High throughput** - Is the use of automated equipment to rapidly test thousands to millions of samples for deployment and unit testing.

WBS



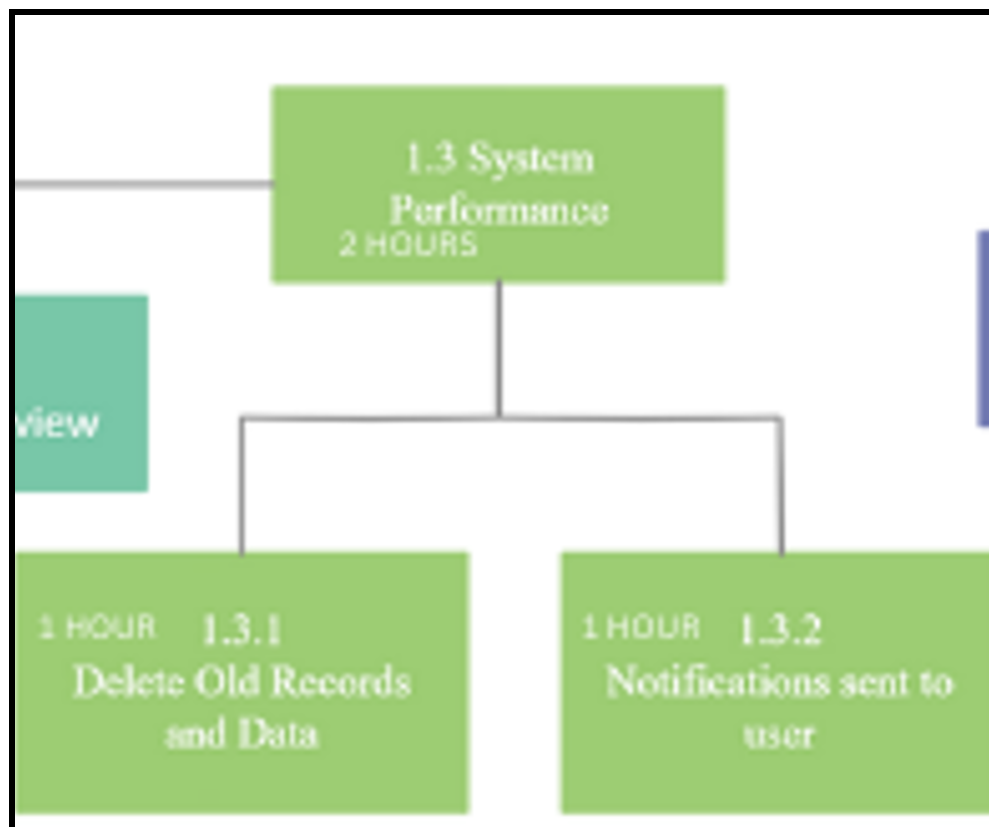
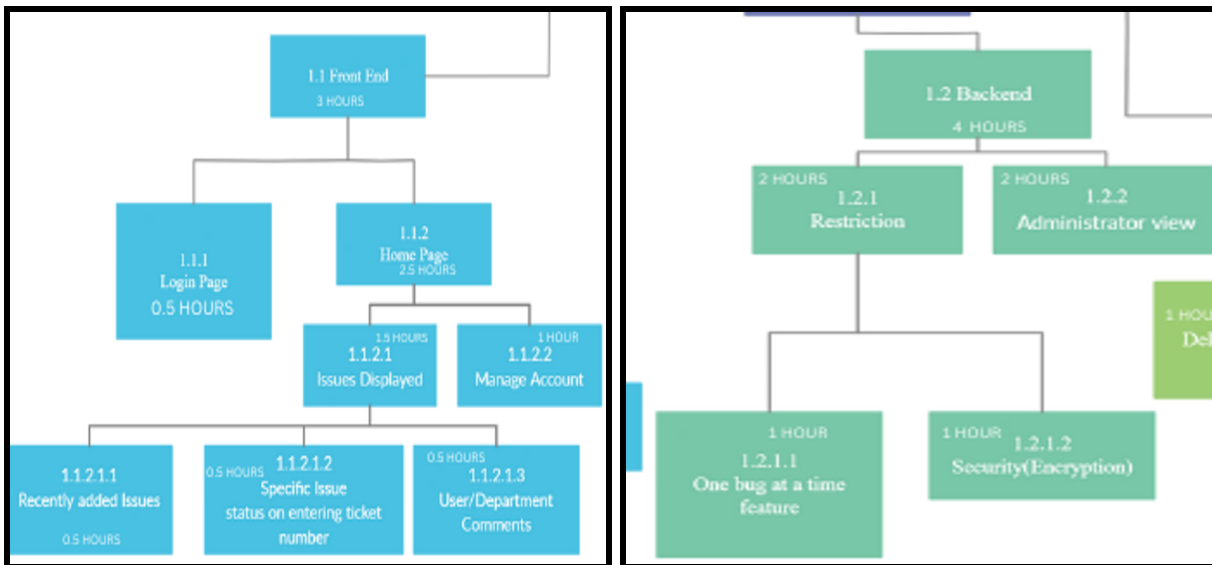
Breaking down the wbs into four principal components, distinguished by the four different colors.

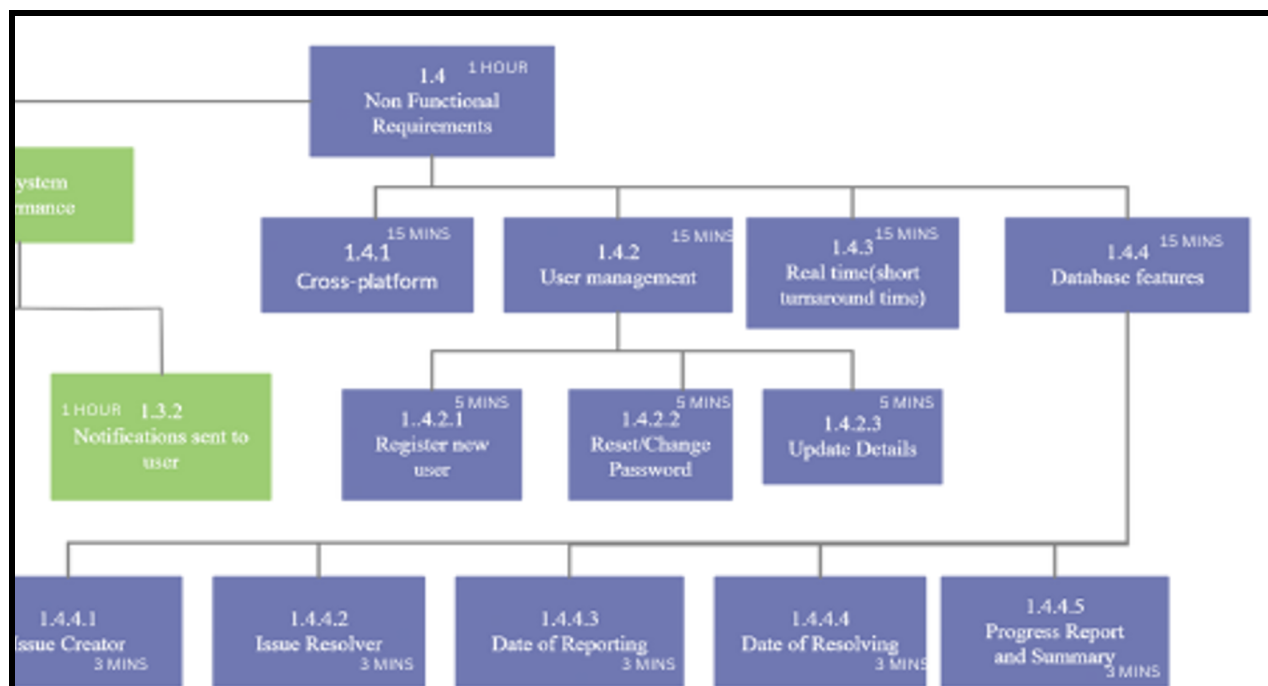
We have successfully broken down the total project task into four phases:

- Front end (3 Hours)
- Back-end (4 Hours)
- System Performance (2 Hours)
- Non Functional(1 Hour)

The systematic breakdown helps us to plan and schedule the tasks in such a manner that we would be able to maintain burndown charts for checking our backlogs, set roadmaps and targets as well as create gantt charts.

Being a very crucial aspect of the project planning phase WBS has a solid structure and a set of rules to be followed. One must be able to clearly identify tasks and subtasks and maintain a hierarchy. For example if a main task is 1.0 and it has two subtasks the subtasks would be 1.1 and 1.2 respectively. insecTo_destrucTo has maintained a well planned WBS and split it's tasks into atomic goals or sub tasks which are clear, concise and unambiguous. Value has been given in scheduling time for each of these tasks and an estimated time of completion has been framed. Finally a gantt chart has been charted to maintain progress of the project during its sprint.





Hourly breakdown has been shown in these individual charts as well to be legibly seen

Estimated time of Completion

Front-end

Login Page - 1 day
Home Page - 4 days
Manage Account - 1 day
Issues Displayed - 3 days
Recently Added Issues - 1 day
Specific Issue status on entering ticket number - 1 day
User/Department comments - 1 day

Back-end

Restriction - 2 days
One bug at a time feature - 1 day
Security (Encryption) - 1 day
Administrator View - 2 days

System Performance

Delete old records and data - 1 day
Notifications sent to user - 1 day

Non-Functional Requirements

Cross-platform - 15 minutes
Real time (short turnaround time) - 15 minutes
User Management - 15 minutes
Register new user - 15 minutes
Reset/Change password - 15 minutes
Update details - 15 minutes
Database features - 15 minutes
Issue Creator - 15 minutes
Issue Resolver - 15 minutes
Date of Reporting - 15 minutes
Date of Resolving - 15 minutes
Progress Report and Summary - 15 minutes

Total Estimated time of Completion : 17 days

Spanning **10 Hours**

GANTT CHART

This is the initial status of the sprint :

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	Start on day	DURATION	PCT OF TASK COMPLETE
1	Front-end						
1.1	Login Page	Anirudh K	17/09/22	18/09/22	0	1	0%
1.2	Home Page	Adarsh L	19/09/22	22/09/22	2	3	0%
1.2.1	Issues Displayed	Ananya A	23/09/22	26/09/22	6	3	0%
1.2.2	Manage Account	Aanchal A	27/09/22	28/09/22	10	1	0%
1.2.1.1	Recently Added Issues	Anirudh K	23/09/22	24/09/22	6	1	0%
1.2.1.2	Specific Issue status on entering ticket number	Adarsh L	24/09/2022	25/09/22	7	1	0%
1.2.1.3	User/Department comments	Ananya A	25/09/22	26/09/22	8	1	0%
2	Back-end						
2.1	Restriction	Aanchal A	27/09/22	29/09/22	10	2	0%
2.2	Administrator View	Anirudh K	30/09/22	01/10/22	13	1	0%
2.1.1	One bug at a time feature	Adarsh L	27/09/22	28/09/22	10	1	0%
2.1.2	Security (Encryption)	Ananya A	28/09/22	29/09/22	11	1	0%
3	System Performance						
3.1	Delete old records and data	Aanchal A	02/10/22	03/10/22	14	1	0%
3.2	Notifications sent to user	Anirudh K	03/10/22	04/10/22	15	1	0%
4	Non-Functional Requirements						
4.1	Cross-platform	Adarsh L	05/10/22	05/10/22	17	0	0%
4.2	User Management	Ananya A	05/10/22	05/10/22	17	0	0%
4.3	Real time (short turnaround time)	Aanchal A	05/10/22	05/10/22	17	0	0%
4.4	Database features	Anirudh K	05/10/22	05/10/22	17	0	0%
4.2.1	Register new user	Adarsh L	05/10/22	05/10/22	17	0	0%

4.2.2	Reset/Change password	Ananya A	05/10/22	05/10/22	17	0	0%
4.2.3	Update details	Aanchal A	05/10/22	05/10/22	17	0	0%
4.4.1	Issue Creator	Anirudh K	05/10/22	05/10/22	17	0	0%
4.4.2	Issue Resolver	Adarsh L	05/10/22	05/10/22	17	0	0%
4.4.3	Date of Reporting	Ananya A	05/10/22	05/10/22	17	0	0%
4.4.4	Date of Resolving	Aanchal A	05/10/22	05/10/22	17	0	0%
4.4.5	Progress Report and Summary	Anirudh K	05/10/22	05/10/22	17	0	0%

