PROJECT REPORT

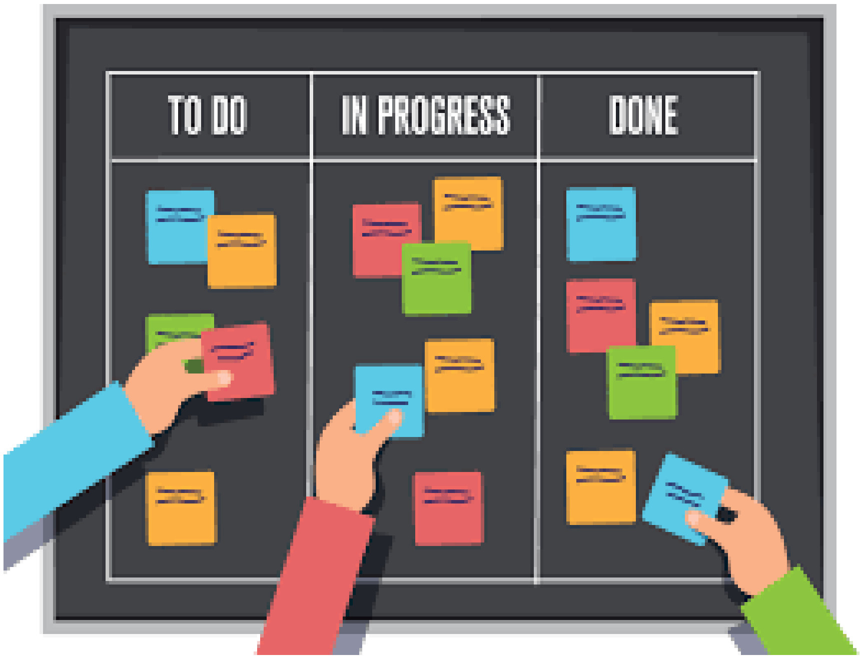
AUTOMATED TOLL MANAGEMENT SYSTEM

TEAM - 11
HRISHIKESH BHAT
B G SOURAV
PRANAV K HEGDE
ADITYA S RAJ

1)IDENTIFYING SOFTWARE LIFE CYCLE MODEL

Due to the nature of the course requirements, a **near AGILE** methodology implementation of **KANBAN** life cycle best suited our project timeline for the following reasons:

- Project building was an **ongoing process** with **regular iterations** as deadlines were to be met with **complying reports and updates**.
- Team members had **no formal roles** and were involved from the stage of inception with full cooperation. Distributed division of labor was based on requirements and not any predefined roles.
- A virtual **Kanban board** or the baseline idea of it was always a part of the project process, right from brainstorming sessions to project delivery and/or submission.
- **Flexibilit**y with regards to **tasks and timings** was issued due to which tasks could be **reprioritized**, **reassigned**, or **update**d as needed



A Simple Kanban Chart

2)WRITING EFFECTIVE USER STORIES

Initially, an **SRS document** was written and documented as per the course requirements. **Revision**s were also made on new findings and understandings but such an implementation is usually a part of **Legacy models**. Hence, an SDLC workflow has **not been implemented** due to the **voracity of work** within the team.

Given below is the **AGILE representation** of the same:

EPIC	USER STORY	ACCEPTANCE CRITERIA
As an **enterprise, we need** to minimize costs, maximize value and provide a service to automate road toll management.	As an **enterprise, we need** to provide a framework of build requirements to achieve automation.	Ensure the **enterprise** is able to: Have a working product that conforms to standards.
As an **end user, I need** to be able to make use of road services with minimal wastage of time and effort.	As an **end user, I need** to pass through a toll gate to pay road taxes and reach my destination.	Ensure the **end user** is able to: Make efficient use of the service with minimal waits.
	As an **end user, I need** to be able to pay the required amount within full legal grounds to avail the service.	
As a **toll plaza admin, I need** to be able to lead local administrative changes in the workflow.	As a **toll plaza admin, I need** to be able to overlook the working and provide local managerial expertise.	Ensure the **admin** is able to: Make toll booth changes in terms of permissions and scalability.
As a **toll booth operator, I need** to be able to process local transactions and maintain automation.	As a **toll booth operator, I need** to be able to ensure smooth traffic flow, successful transactions and handle real time exceptions.	Ensure the operator is able to: Perform basic tool operations with automation at transaction level.

3)<u>SOFTWARE PROTOTYPING</u>

Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation.

- Our prototype was based on using the local Mysql database flavor, MariaDB.
- Simple SQL data definition was used to create and populate the database.
- Since we followed database building iteratively, right from identifying entities and relations to altering constraints, normalization is not required at every step.
- Regular SQL queries were used to check the integrity of the database including references, derivations and constraints.
- Now, a user-friendly and simple GUI has been implemented to avoid direct querying and provide abstraction for the same.
- Also, multiple iterations over how privileges will be set have been discussed such as direct database access or interface only access etc and the most suitable method has been implemented.

Due to this methodology of prototyping**,** **quicker feedback** was available**,** missing **functionalities** were identified and **re-use** was established.

4)WORK BREAKDOWN STRUCTURE

JIRA as a service was extensively used to create **Epics, Stories** and **Tasks** with respect to the project.

In the **Roadmap** section, a fellow **assignee** would create Epics with relevant stories and tasks. A teammate would be assigned to a task based on immediate requirements.

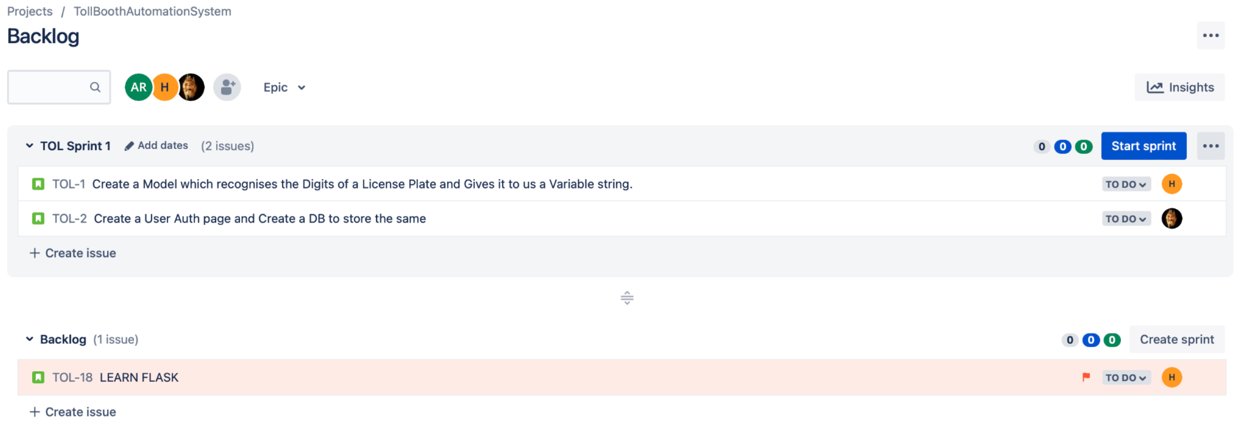
A task can be then marked based on the level of completion as **IN PROGRESS** or **DONE.**

An initial **SPRINT** over a period of two weeks was established to implement certain functionalities.

Any functionality that was outside the immediate scope of implementation was identified and **flagged** under **Backlogs.**

Code can either be linked directly to JIRA via GitHub but since exploring GitHub was a task by itself, implementation has been done independently.

Sprint and Backlog creation using JIRA



TOL Sprint 1 Add dates (2 issues) 0 0 0

TOL-1 Create a Model which recognises the Digits of a License Plate and Gives it to us a Variable string. TO DO H

TOL-2 Create a User Auth page and Create a DB to store the same TO DO

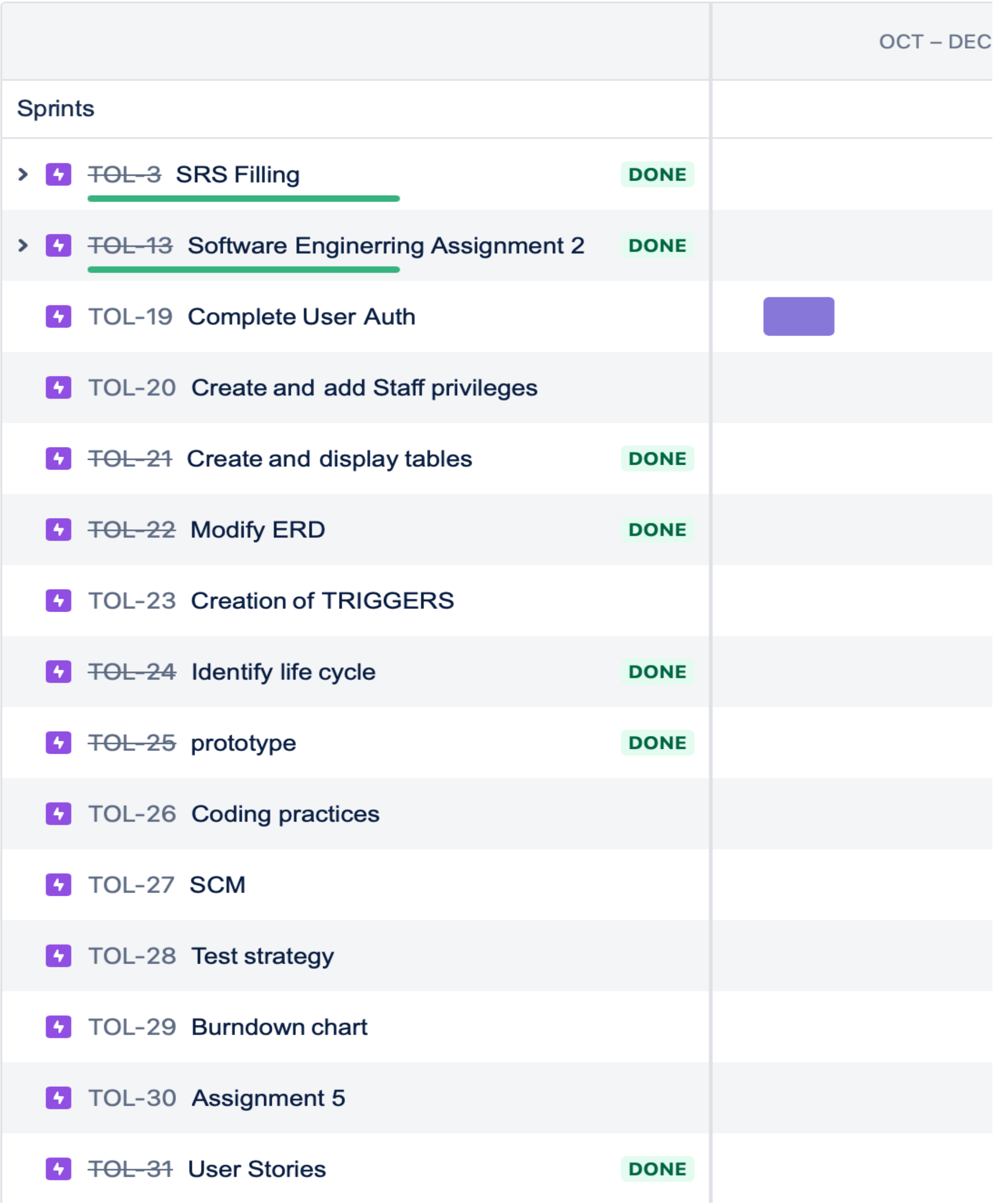
+ Create issue

Backlog (1 issue) 0 0 0

TOL-18 LEARN FLASK TO DO H

+ Create issue

Work Breakdown Structure using JIRA

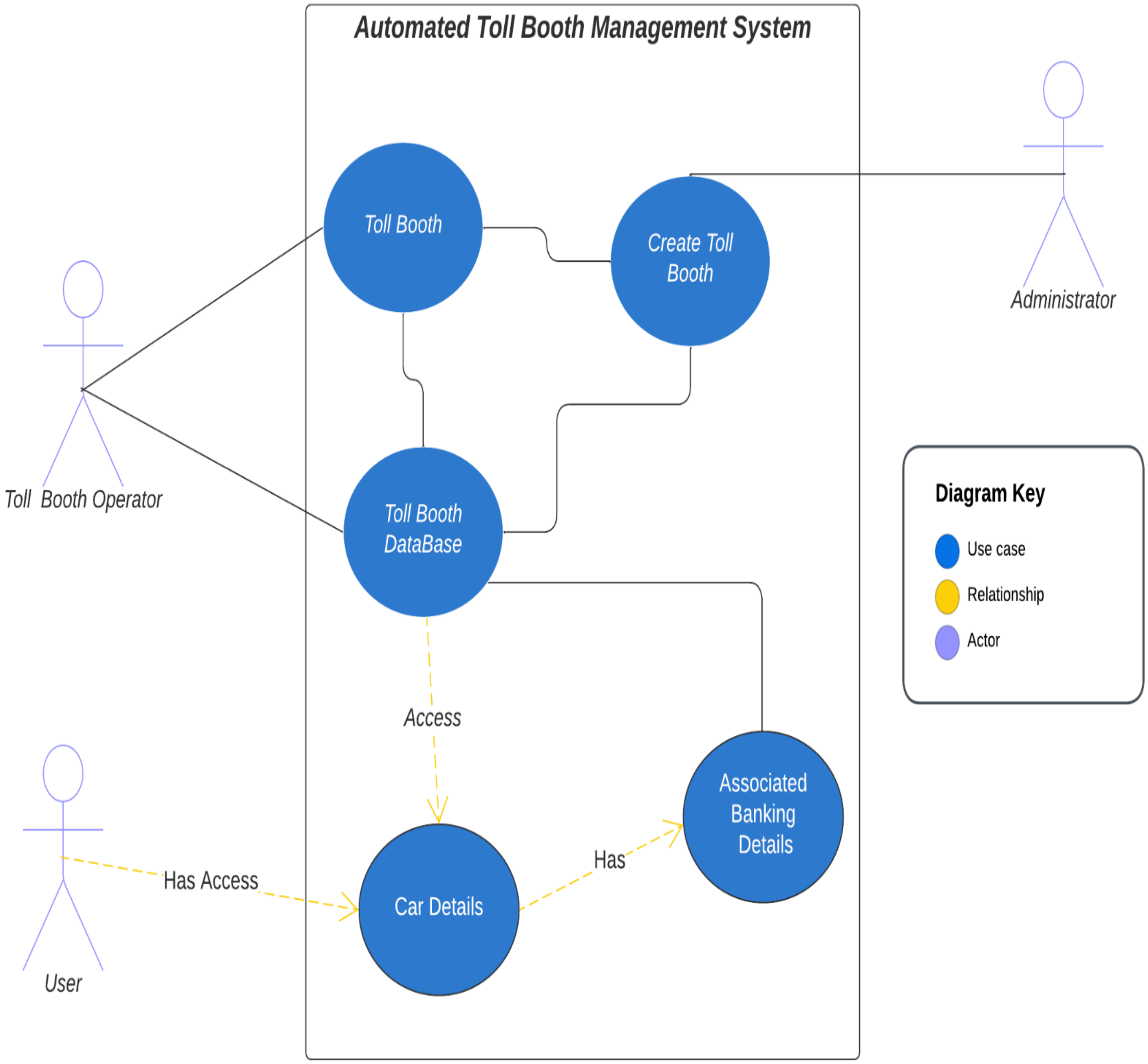


			OCT – DEC
Sprints			
~~TOL-3~~	SRS Filling	DONE	
~~TOL-13~~	Software Enginerring Assignment 2	DONE	
TOL-19	Complete User Auth		▬
TOL-20	Create and add Staff privileges		
~~TOL-21~~	Create and display tables	DONE	
~~TOL-22~~	Modify ERD	DONE	
TOL-23	Creation of TRIGGERS		
~~TOL-24~~	Identify life cycle	DONE	
~~TOL-25~~	prototype	DONE	
TOL-26	Coding practices		
TOL-27	SCM		
TOL-28	Test strategy		
TOL-29	Burndown chart		
TOL-30	Assignment 5		
~~TOL-31~~	User Stories	DONE	

5)PROJECT DESIGN

Unified modeling language (UML) diagrams are designed to help clarify project requirements at the **front end of agile development** by providing a **visual understanding** of how a project should **look and function**.

Since **Kanban** follows an **AGILE framework**, a concise UML diagram should provide maximum coverage for the same.
To make things simpler, a **Use-Case diagram is a behavioral UML diagram** while an **Architecture diagram is not recommended** to be used for AGILE implementation.



6)CODING PRACTICES AND STANDARDS

Coding standards are a set of rules, techniques, and best practices to create cleaner, more readable, more efficient code with minimal errors. They offer a uniform format using which we can use to build sophisticated and highly functional code.

This has to be implemented in order to achieve the following:

- Uniformity to the code created by different engineers.
- Creation of reusable code.
- Easier to detect errors.
- Make code simpler, more readable, and easier to maintain.
- Boost programmer efficiency and generate faster results.

Since we have used tools and languages that conform to industry standards, uniformity can be checked off the list.

Code readability is maintained by using precise function names, short lines of code, proper indentation, shallow nesting and segmentation.

Standardization of module names is done for readability.

Daily or very regular backups through branching, merging and version management is implemented.

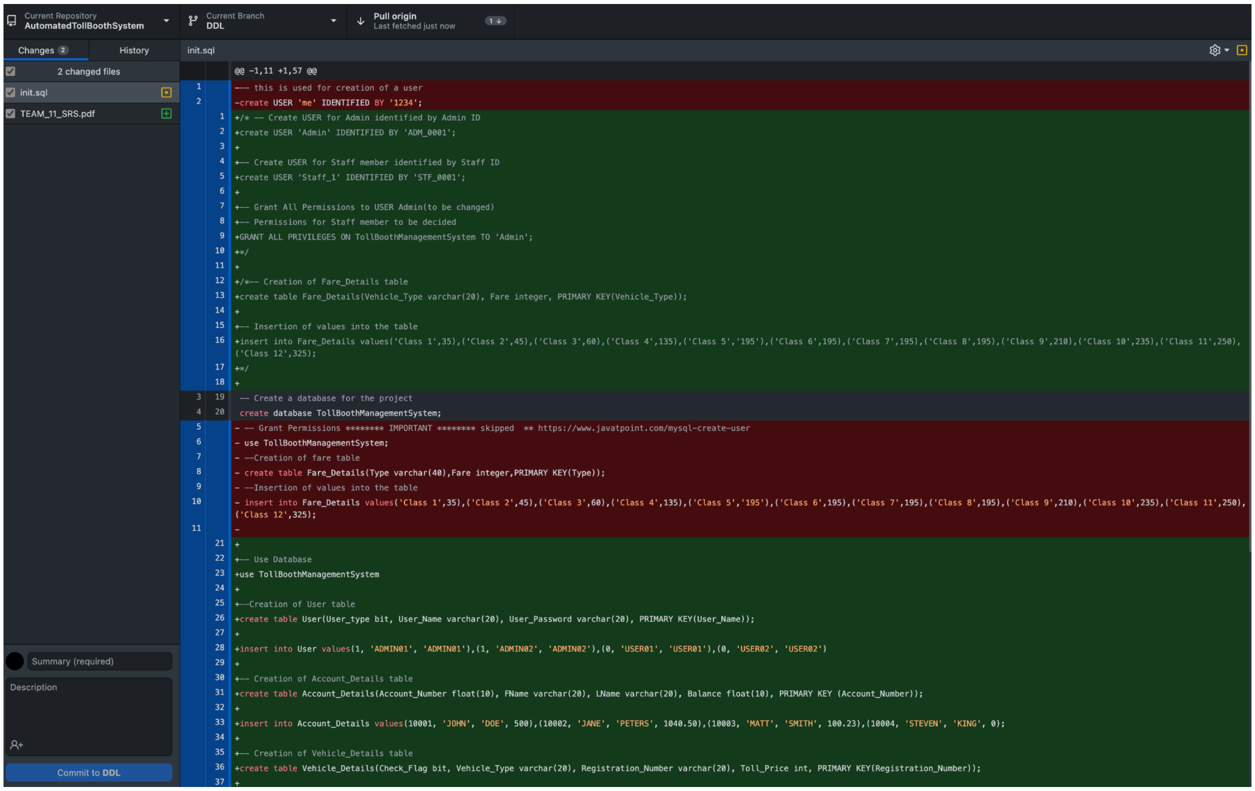
Exception Handling has been formalized for log reads and minimum critical damage

7)SOFTWARE CHANGE MANAGEMENT

Change management in software development involves tracking and managing changes to artifacts, such as code and requirements. It's critical for effective application development.

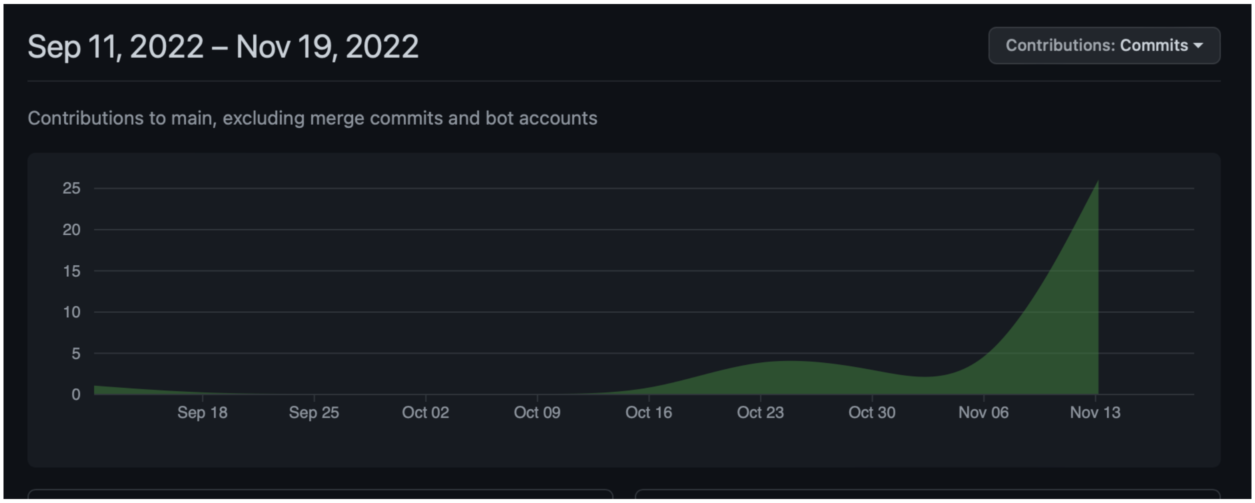
GitHub and GitHub desktop were extensively used for authoritative commits.
Code changes outweighed the changes in requirements and any requirement changes were just revisions of the documentation.

We will be providing insights, commit history and version maintenance through the GitHub interface. It is to be noted that there is a provision in JIRA to connect and work with GitHub.



```
@@ -1,11 +1,57 @@
--- this is used for creation of a user
-create USER 'me' IDENTIFIED BY '1234';
+/* -- Create USER for Admin identified by Admin ID
+create USER 'Admin' IDENTIFIED BY 'ADM_0001';
+
+-- Create USER for Staff member identified by Staff ID
+create USER 'Staff_1' IDENTIFIED BY 'STF_0001';
+
+-- Grant All Permissions to USER Admin(to be changed)
+-- Permissions for Staff member to be decided
+GRANT ALL PRIVILEGES ON TollBoothManagementSystem TO 'Admin';
+*/
+
+/*-- Creation of Fare_Details table
+create table Fare_Details(Vehicle_Type varchar(20), Fare integer, PRIMARY KEY(Vehicle_Type));
+
+-- Insertion of values into the table
+insert into Fare_Details values('Class 1',35),('Class 2',45),('Class 3',60),('Class 4',135),('Class 5','195'),('Class 6',195),('Class 7',195),('Class 8',195),('Class 9',210),('Class 10',235),('Class 11',250),
('Class 12',325);
+*/
+
 -- Create a database for the project
 create database TollBoothManagementSystem;
- -- Grant Permissions ******** IMPORTANT ******** skipped  ** https://www.javatpoint.com/mysql-create-user
- use TollBoothManagementSystem;
- --Creation of fare table
- create table Fare_Details(Type varchar(40),Fare integer,PRIMARY KEY(Type));
- --Insertion of values into the table
- insert into Fare_Details values('Class 1',35),('Class 2',45),('Class 3',60),('Class 4',135),('Class 5','195'),('Class 6',195),('Class 7',195),('Class 8',195),('Class 9',210),('Class 10',235),('Class 11',250),
('Class 12',325);
-
+
+-- Use Database
+use TollBoothManagementSystem
+
+--Creation of User table
+create table User(User_type bit, User_Name varchar(20), User_Password varchar(20), PRIMARY KEY(User_Name));
+
+insert into User values(1, 'ADMIN01', 'ADMIN01'),(1, 'ADMIN02', 'ADMIN02'),(0, 'USER01', 'USER01'),(0, 'USER02', 'USER02')
+
+-- Creation of Account_Details table
+create table Account_Details(Account_Number float(10), FName varchar(20), LName varchar(20), Balance float(10), PRIMARY KEY (Account_Number));
+
+insert into Account_Details values(10001, 'JOHN', 'DOE', 500),(10002, 'JANE', 'PETERS', 1040.50),(10003, 'MATT', 'SMITH', 100.23),(10004, 'STEVEN', 'KING', 0);
+
+-- Creation of Vehicle_Details table
+create table Vehicle_Details(Check_Flag bit, Vehicle_Type varchar(20), Registration_Number varchar(20), Toll_Price int, PRIMARY KEY(Registration_Number));
+
```

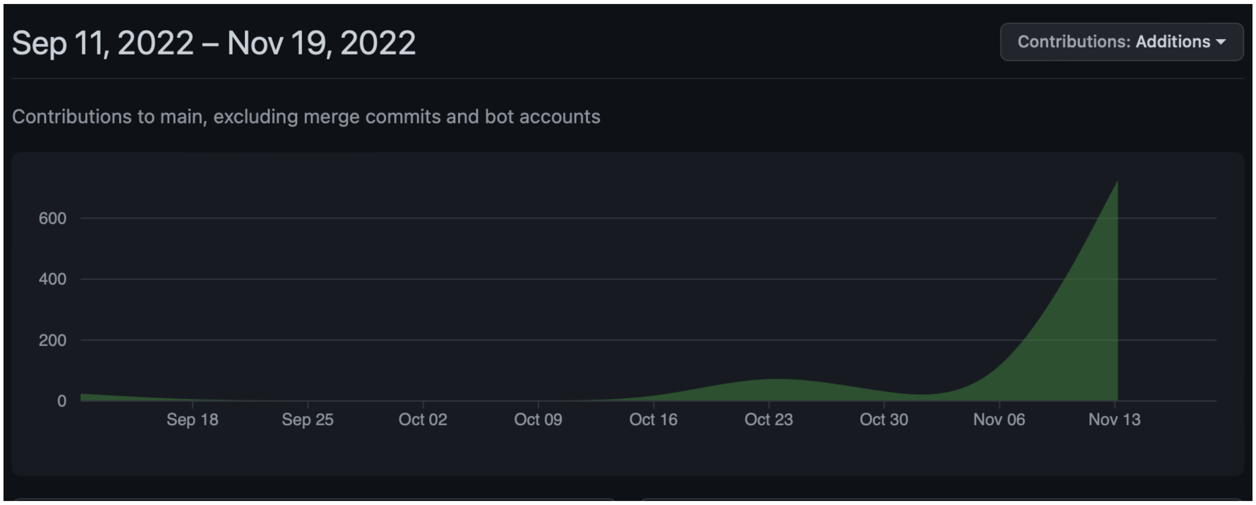
GitHub Desktop interface for commits and merges



Sep 11, 2022 – Nov 19, 2022

Contributions to main, excluding merge commits and bot accounts

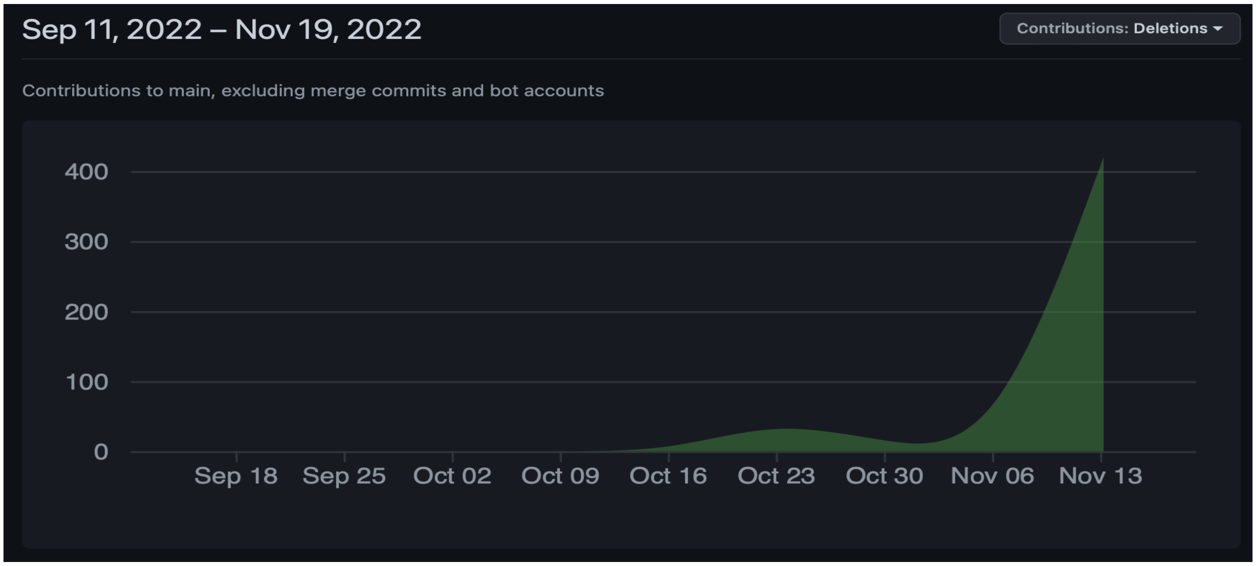
Commit frequency



Sep 11, 2022 – Nov 19, 2022

Contributions to main, excluding merge commits and bot accounts

Addition frequency

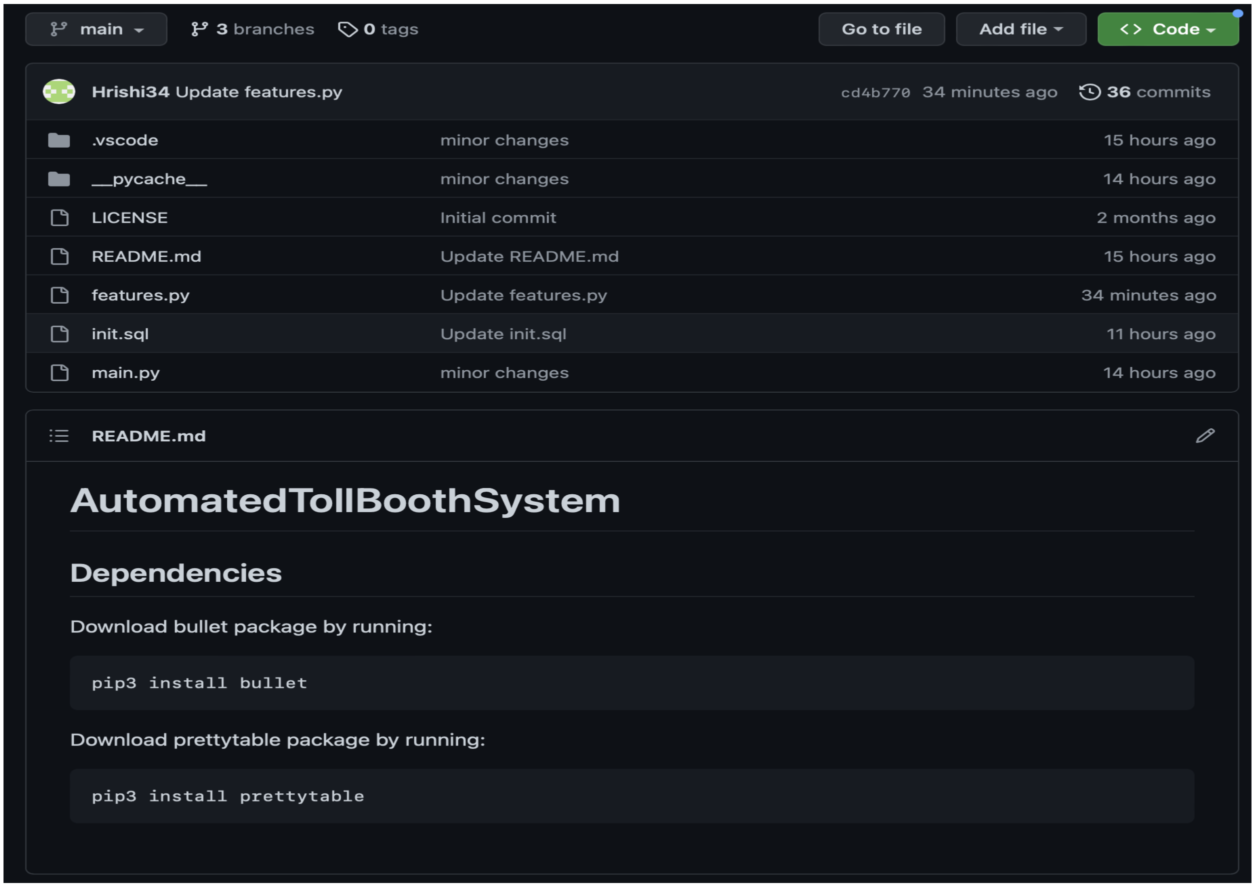


Sep 11, 2022 – Nov 19, 2022

Contributions: Deletions

Contributions to main, excluding merge commits and bot accounts

Deletion frequency



Hrishi34 Update features.py cd4b770 34 minutes ago 36 commits

.vscode	minor changes	15 hours ago
__pycache__	minor changes	14 hours ago
LICENSE	Initial commit	2 months ago
README.md	Update README.md	15 hours ago
features.py	Update features.py	34 minutes ago
init.sql	Update init.sql	11 hours ago
main.py	minor changes	14 hours ago

README.md

AutomatedTollBoothSystem

Dependencies

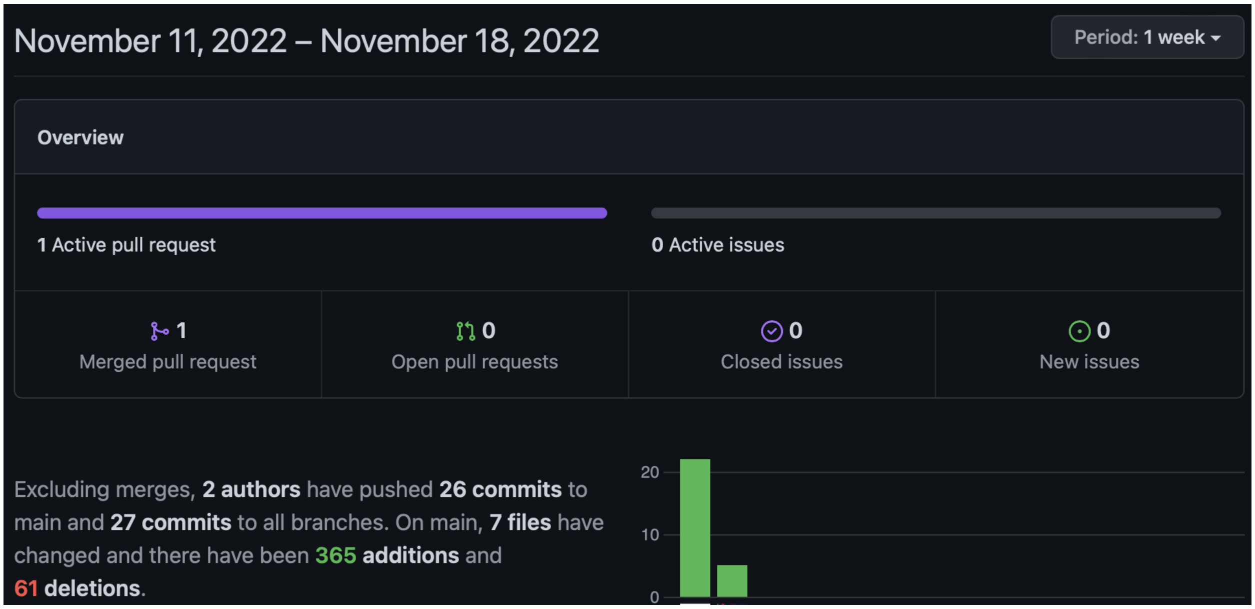
Download bullet package by running:

```
pip3 install bullet
```

Download prettytable package by running:

```
pip3 install prettytable
```

Project Files



November 11, 2022 – November 18, 2022

Overview

1 Active pull request

0 Active issues

1 Merged pull request

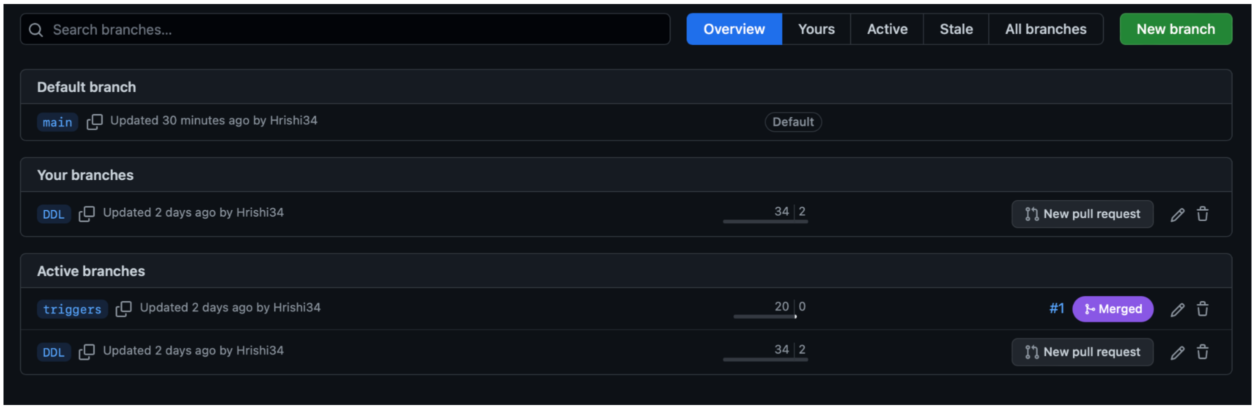
0 Open pull requests

0 Closed issues

0 New issues

Excluding merges, **2 authors** have pushed **26 commits** to main and **27 commits** to all branches. On main, **7 files** have changed and there have been **365 additions** and **61 deletions**.

Latest Project Overview



Default branch

main Updated 30 minutes ago by Hrishi34 Default

Your branches

DDL Updated 2 days ago by Hrishi34 34 | 2 New pull request

Active branches

triggers Updated 2 days ago by Hrishi34 20 | 0 #1 Merged

DDL Updated 2 days ago by Hrishi34 34 | 2 New pull request

Branch and Merge history

8)PROJECT TEST STRATEGY

TEST PLAN:

Typical testing lifecycle is **not followed** due to the **KANBAN implementation of AGILE** in this project. Kanban focuses on a visual-driven model for project management with continuous improvement and transparency in workflows. At the simplest level it is all about **visualizing project status** to clearly understand what's sitting in a backlog waiting to get picked up, what work is currently in progress and what work is complete.

Few general testing steps, such as **"Upcoming", "In Progress" and "Complete,**" or more function-specific, such as **"Test Plan", "Test Design", "Test Execution" and "Test Reporting"** are defined for the team's working clarity.

Next, we created **cards for each task**. As work begins, we move these cards through each step in the workflow to **map their progress** and make it easy for everyone to identify where each project or task stands at any given time.
These are followed for each **necessary module with valid and invalid test cases** in the form of a **test suite**.

With this approach, we ensure to enable the following:

- Speedy resolution of bottlenecks
- Better project commitment
- Improved planning and prioritization
- Faster cycle times

TEST SUITE:

Module: Function “**login**”

Test case id	Action	Inputs	Expected Output	Actual output	Test result	Test comments
TS100	Enter a valid admin/staff username and password	[Admin] Username :‘ADMIN01’ Password: ‘ADMIN1’	“Logged in as Admin..”	“Logged in as Admin..”	PASSED	Valid credentials logs in the user into the program and displays Logged in message to confirm
TS101	Enter invalid admin/staff username or password	[Staff] Username: ‘Asdflkj’ Password: ‘Staff1’	“Error occurred: 1045 (28000): Access denied for user . Credentials do not match”	“Error occurred: 1045 (28000): Access denied for user Asdflkj’@'localhost' (using password: YES) Credentials do not match”	PASSED	Invalid credentials displays mysql.connector error to the the user. Attempt to login stopped and will be asked to try again.
TS102	Enter username with length less than 4 characters.	Username: ‘ISA’ Password: ‘ADMIN1’	“Either USERNAME does not exist or invalid PASSWORD Login failed! “	“Either USERNAME does not exist or invalid PASSWORD Login failed! “	PASSED	The program requires users to have username length to be minimum 4 characters
TS103	Enter username as ‘root’ and any password	Username: ‘root’ Password: ‘ADMIN1’	“Either USERNAME does not exist or invalid PASSWORD Login failed! ”	“Either USERNAME does not exist or invalid PASSWORD Login failed! ”	PASSED	Since ‘root’ has all privileges to the database, the user is not allowed to login as root, not even Admin.

Module :Function “**Car_entered**”

Test case id	Action	Inputs	Expected output	Actual output	Test result	Test comments
TS200	Enter a Vehicle register number which is present in the database.	Register number: ‘CH6540’	‘Vehicle number is present in the database.	‘Vehicle is present in database’	PASSED	If the register number is present in the database then the car_present function is called.
TS201	Enter a Vehicle register number which is not present in the database.	Register number: ‘KA1234’	Vehicle number is not present in the database.	‘Vehicle is not present in the database. Please register.’	PASSED	If the register number is not present in the database then the car_notpresent function is called to register into the database.

Module: Function”**Create_Staff**”

Test case id	Action	Inputs	Expected output	Actual output	Test result	Test comments
TS300	Create a new staff user	Username:Staff_09 Password:Staff_09	User has been successfully Created	User has been successfully Created	PASSED	If the username which is entered is a unique username the flow passes to the next state
TS301	Try to create a new staff with the existing staff’s name.	Username:Staff_09 Password:Staff_09	Error has occurred The staff already exists	Error has occurred The staff already exists	PASSED	If the username which is meant to be created already exists in the database then the error prompt has to be shown

Module: Function”**main.py**”

Test case id	Action	Inputs	Expected output	Actual output	Test result	Test comments
TS400	Enter a valid choice in the prompt [‘Y’/’N’]	‘Y’, ‘N’	Prompt to Login “Enter username (Minimum length is 4):”	Prompt to Login “Enter username (Minimum length is 4):”	Passed	The code expects ‘Y’or ‘N’ as inputs. So we check by giving them as inputs and checking the behavior of the program
TS401	Enter an invalid choice in the prompt [‘Y’/’N’]	‘U’,’i’,’0’,’-’	“Please enter a valid choice”	“Please enter a valid choice” And reprompt	Passed	Enter an Invalid input to check the robustness of the program in handling such choices
TS402	Enter the valid choice in lowercase	‘y’,’n’	Prompt to Login “Enter username (Minimum length is 4):”	Prompt to Login “Enter username (Minimum length is 4):”	Passed	Enter the same choices in lower case to ensure that the program runs as the user intends

Module: Function “**low_balance**”

Test Case Id	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments

TS500	Entering the minimum amount or more required for the recharge.	Recharge: {amount}	Recharge is successful.	Recharge is successful.	PASSED	When the user enters the amount required or more than that for the toll tax.
TS501	Entering the amount lesser than the minimum amount specified.	Recharge: {amount}	Please enter minimum amount	Please enter minimum amount	PASSED	When the user enters or recharges with the amount lesser than the specified minimum amount.

Module: Function “**enough_balance**”

Test Case Id	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
TS600	After deducting the required amount, Recharging the account.	Recharge ch: {Y/N}: Y	Enter the recharge amount (Recommended if Balance is less than Tax):	Enter the recharge amount (Recommended if Balance is less than Tax):	PASSED	User wants to recharge the amount available in his account.

TS601	The Recharge amount given is invalid and given alert for the same.	Recharge : {invalid amount}	Invalid amount	Invalid amount	PASSED	When the user gives the amount in some invalid fashion such as a number less than zero.
TS602	The Recharge amount is a valid amount for future use.	Recharge : {valid amount}	Recharge is successful.	Recharge is successful.	PASSED	Shows the prompt for successful recharge and prints new balance too.

Module: Function “**update_chkbit**”

Test Case Id	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
TS700	Updating the check bit value to 1.	Balance	Check bit updated.	Check bit updated.	PASSED	When the balance given is enough or greater than the price of the toll tax.
TS701	Updating the check bit value to 0.	Balance	Check bit updated.	Check bit updated.	PASSED	When the balance given is low or lesser than the price of the toll tax.

Module: Function “**car_notpresent**”

Test Case Id	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
TS800	Enter a valid option for the Type of Vehicle.	Vehicle Type: 4, 3, 2 or 1	You have chosen Vehicle Type as – {Vehicle Type}	You have chosen Vehicle Type as – {Vehicle Type}	PASSED	Entering a valid option for the vehicle details and saving the type of vehicle passing through the booth.
TS801	Checking and validating errors in the Information of the personal details and inserting in the database.	First Name, Last Name, Account Number, Balance in the Account.	Error has Occurred!! i.e.: {e}	Error has Occurred!! i.e.: {e}	PASSED	If the given information is incorrect or incompatible, then the data is inserted into the Database as given.
TS802	If the Vehicle is a Government Vehicle, then give it a free pass.	Check Flag: 2 or 1	This is a government vehicle and is exempted from tax.	This is a government vehicle and is exempted from tax.	PASSED	Government Vehicles are given a free pass.

TS803	If the non-government vehicle has low balance, then prompts low balance.	Check flag: 1	Not sufficient balance!	Not sufficient balance!	PASSED	The user must pay the price as toll tax and the prompt for low balance is as shown and keeps asking till the user recharges.

Module : Function “**Car_present**”

Test Case Id	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
TS900	Enter invalid toll booth number	Toll booth number: 10	‘Toll booth number exceeded.Try again’	‘Please enter the toll booth number where the vehicle entered [less than 4]’	PASSED	If the entered toll booth number does not exist , the user is asked to enter a valid tool booth number again.
TS901	[Balance lesser than fare] Enter valid toll booth number,account number choice,recharge amount if low balance,transaction id.	Toll booth number: 2, Account choice: 1, Recharge amount: 200, Transaction id=’TR100201’	Successful recharge and tax collection	‘Recharge successful.. Tax deducted successfully.. Remaining balance : <Amount>. Toll Booth Revenue updated.’	PASSED	If the recharge is done successfully then the amount balance will be updated after deduction of tax.

TS902	[Balance greater than fare] Enter valid toll booth number,account number,Choice to recharge.	Toll booth number: 2, Account choice: 1 Recharge choice: N	Tax deducted successfully and revenue of the particular toll booth updated .	‘Tax deducted successfully.. Remaining balance : <Amount>, Toll Booth Revenue updated.’	PASSED	If the user has sufficient balance the tax is deducted automatically and the user is asked if he wants to recharge .

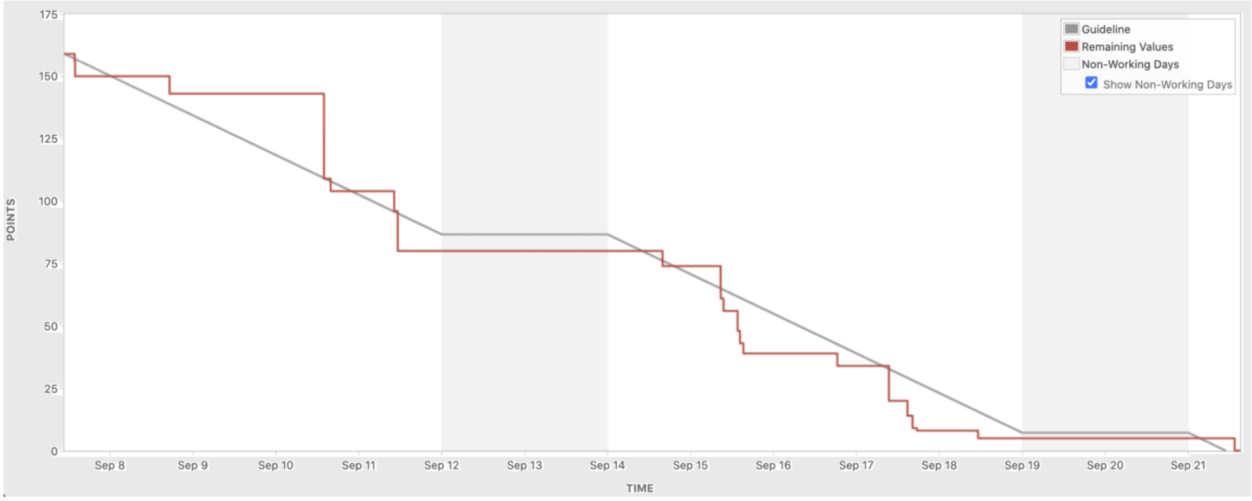
Module: Function “**Transac_id**”

Test Case Id	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
TS1000	Enter valid transaction number	Transaction number: TR220000	‘Successfully tax deducted’	‘Recharge successful.. Tax deducted successfully.. ’	PASSED	If the transaction id is valid then it is inserted into the transaction table and function returns.
TS1001	Enter repeated/invalid transaction number	Transaction number: TR220022	‘Enter a valid transaction id.’	‘Error occurred : Please try again.. ’	PASSED	If the entered toll booth number is repeated , the user is asked to enter a valid id.

9)BURNDOWN CHART

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time

A burndown report has been generated for the very first sprint that included documentation and analysis with a suitable deadline.



Additional benefits of using a burndown chart include:

1. **Shows a direct comparison:** A burndown chart shows a direct comparison between the work needed to be done and the effort needed to complete the sprint. This helps us connect tasks to larger goals and can keep tasks moving on pace with sprint goals.
2. **Keeps teams on the same page**: With a daily effort log and a place to visualize the work needed, team members have one source of information that they can track and connect to about the tasks at hand.
3. **Gives insight into team productivity:** Not only is a burndown chart great at visualizing work, but it can also give us insight into how productive our team is and how quickly we work. If our actual work is drastically different from our ideal, then we can work on helping our team be more productive.

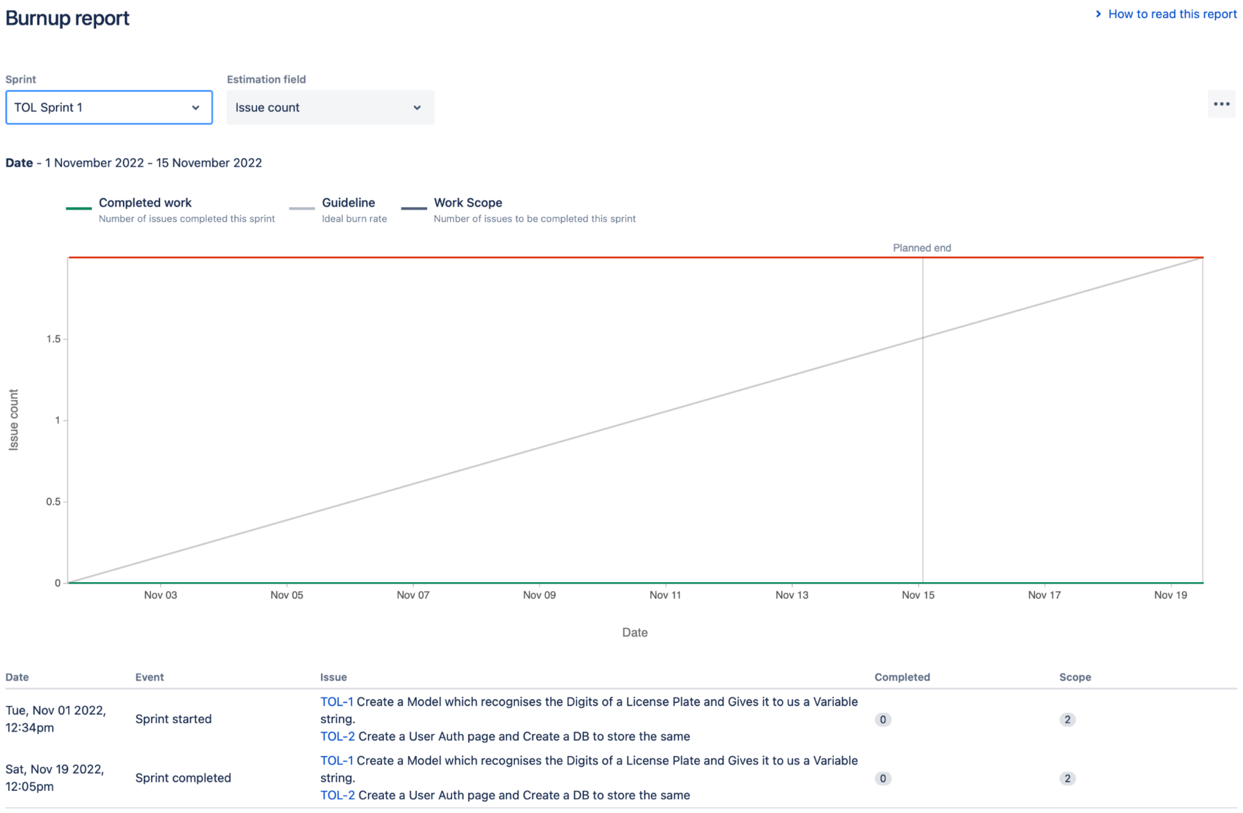
Based on the very first Burndown chart, any fields that our team missed out on or could show major improvements in was looked into for the next few sprints

10)JIRA REPORT

Due to the absence of an active SCRUM master and presence of team management, there are only two major reports

Burnup Chart

Applies to: Sprints
Provides a visual representation of a sprint's scope, as well as its remaining work.
This helps your team stay on track.



Burnup report

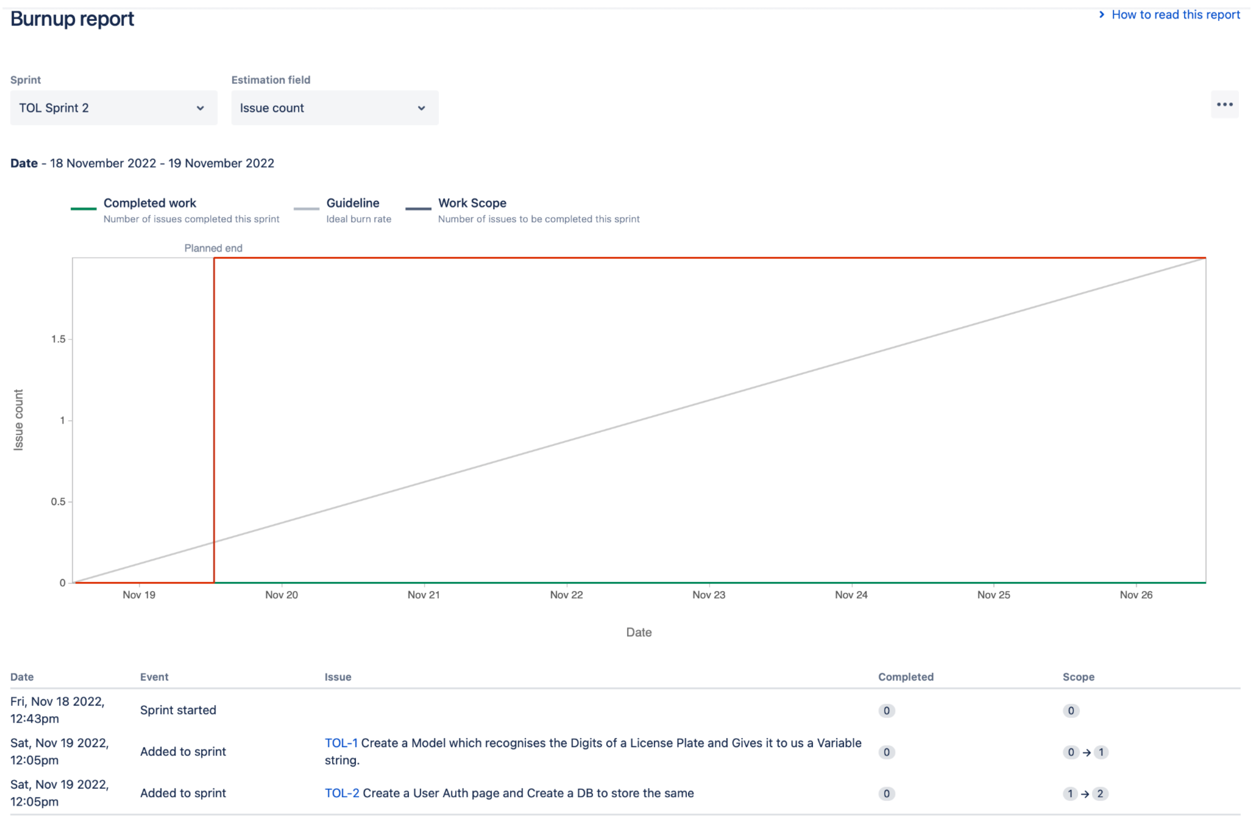
› How to read this report

Sprint: TOL Sprint 1

Estimation field: Issue count

Date - 1 November 2022 - 15 November 2022

Date	Event	Issue	Completed	Scope
Tue, Nov 01 2022, 12:34pm	Sprint started	TOL-1 Create a Model which recognises the Digits of a License Plate and Gives it to us a Variable string. TOL-2 Create a User Auth page and Create a DB to store the same	0	2
Sat, Nov 19 2022, 12:05pm	Sprint completed	TOL-1 Create a Model which recognises the Digits of a License Plate and Gives it to us a Variable string. TOL-2 Create a User Auth page and Create a DB to store the same	0	2



Burnup report

How to read this report

Sprint: TOL Sprint 2

Estimation field: Issue count

Date - 18 November 2022 - 19 November 2022

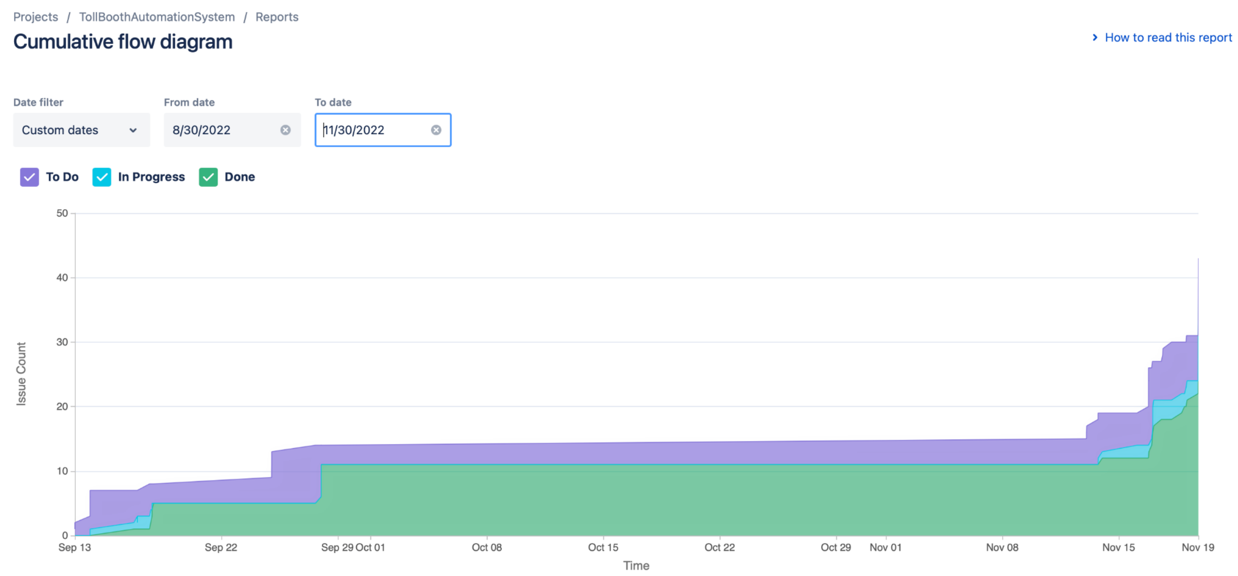
Date	Event	Issue	Completed	Scope
Fri, Nov 18 2022, 12:43pm	Sprint started		0	0
Sat, Nov 19 2022, 12:05pm	Added to sprint	TOL-1 Create a Model which recognises the Digits of a License Plate and Gives it to us a Variable string.	0	0 → 1
Sat, Nov 19 2022, 12:05pm	Added to sprint	TOL-2 Create a User Auth page and Create a DB to store the same	0	1 → 2

Velocity Chart

Applies to: Sprints
Tracks the amount of work completed from sprint to sprint.
This helps you determine your team's velocity, and estimate the work your team can realistically achieve in future sprints.

Cumulative Frequency

Applies to: Sprints
Tracks the progress status of all assigned tasks over a timeline



Projects / TollBoothAutomationSystem / Reports

Cumulative flow diagram

› How to read this report

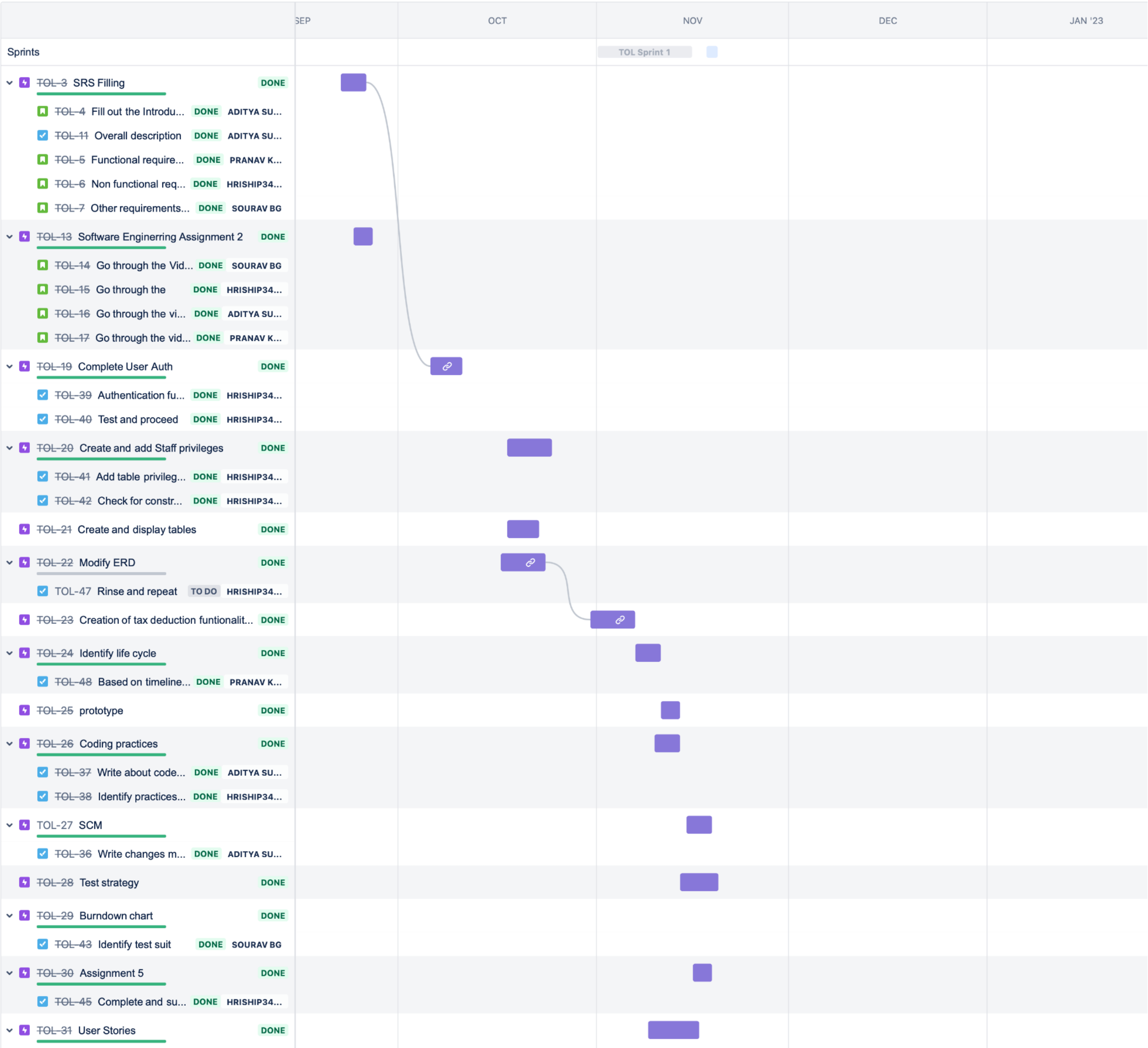
Date filter: Custom dates

From date: 8/30/2022

To date: 11/30/2022

☑ To Do ☑ In Progress ☑ Done

For a broader perspective, a near complete Roadmap report will serve the purpose including creation, status and deadline timelines with multiple assignee and assigned members.



Issue	Summary	Status	Assignee
~~TOL-3~~	SRS Filling	DONE	
~~TOL-4~~	Fill out the Introdu...	DONE	ADITYA SU...
~~TOL-11~~	Overall description	DONE	ADITYA SU...
~~TOL-5~~	Functional require...	DONE	PRANAV K...
~~TOL-6~~	Non functional req...	DONE	HRISHIP34...
~~TOL-7~~	Other requirements...	DONE	SOURAV BG
~~TOL-13~~	Software Enginerring Assignment 2	DONE	
~~TOL-14~~	Go through the Vid...	DONE	SOURAV BG
~~TOL-15~~	Go through the	DONE	HRISHIP34...
~~TOL-16~~	Go through the vi...	DONE	ADITYA SU...
~~TOL-17~~	Go through the vid...	DONE	PRANAV K...
~~TOL-19~~	Complete User Auth	DONE	
~~TOL-39~~	Authentication fu...	DONE	HRISHIP34...
~~TOL-40~~	Test and proceed	DONE	HRISHIP34...
~~TOL-20~~	Create and add Staff privileges	DONE	
~~TOL-41~~	Add table privileg...	DONE	HRISHIP34...
~~TOL-42~~	Check for constr...	DONE	HRISHIP34...
~~TOL-21~~	Create and display tables	DONE	
~~TOL-22~~	Modify ERD	DONE	
TOL-47	Rinse and repeat	TO DO	HRISHIP34...
~~TOL-23~~	Creation of tax deduction funtionalit...	DONE	
~~TOL-24~~	Identify life cycle	DONE	
~~TOL-48~~	Based on timeline...	DONE	PRANAV K...
~~TOL-25~~	prototype	DONE	
~~TOL-26~~	Coding practices	DONE	
~~TOL-37~~	Write about code...	DONE	ADITYA SU...
~~TOL-38~~	Identify practices...	DONE	HRISHIP34...
TOL-27	SCM	DONE	
~~TOL-36~~	Write changes m...	DONE	ADITYA SU...
~~TOL-28~~	Test strategy	DONE	
~~TOL-29~~	Burndown chart	DONE	
~~TOL-43~~	Identify test suit	DONE	SOURAV BG
~~TOL-30~~	Assignment 5	DONE	
~~TOL-45~~	Complete and su...	DONE	HRISHIP34...
~~TOL-31~~	User Stories	DONE	