Bug Tracking System

BugVITa

Software Design Specification

Version: 1.0

Date Created: 2021.04.17

Signatures

Date	Revision	Approved By
2021.03.23	1.0	Aashish Sharma
2021.03.23	1.0	Priyanshu Baban Gaikwad
2021.03.23	1.0	Srikanth Balakrishna

List of Contributors

Name	Initials	E-Mail	Reg. No.
Aashish Sharma	AS	aashish.sharma2019@vitstudent.ac.in	19BCE0971
Priyanshu Gaikwad	PG	priyanshu.gaikwad2019@vitstudent.ac.in	19BCE0550
Srikanth Balakrishna	SB	srikanth.balakrishna2019@vitstudent.ac.in	19BCE0158

Change History

Revision	Date	Description
1.0	2021.04.17	Initial version

Preface

This document presents the Software Design Specification for BugVITa. The major sections of the document address the system decomposition by module, concurrent process, and data entity. The system dependencies are also described.

Section 2, Decomposition Description, gives a view of the whole system design including concurrent processes and data entities that are common amongst all system modules. This discussion includes a UML Class Diagram that depicts the entire system.

Section 4, Interface Description, goes into detail about the user interface for each module of BugVITa. This is followed by an important discussion of the processes implemented in logic for each module of the system.

Section 5, Detailed Design, extends the design discussion found in Section 2 and describes the design for each system module in more detail. A UML Class diagram is included for each module design discussion. This is followed by a description of the data requirements for each module and the design of those data elements.

Table of Contents

List of Tables

1. Introduction

Purpose

The purpose of the Software Design Specification is to describe the specific design of BugVITa. The design specification includes an overview of the design along with software module decomposition.

This document provides a detailed description of each software module's design. For each module, a user interface design and class diagram design is given. As well, a process description is described for each module. It is in the process description that the details of what logic will need to be implemented are given.

Scope

It is within the scope of the Software Design Specification to describe the specific system design of BugVITa. This would include user interface design, object-oriented class design, process design, and data design. Any specific detail that is needed about the standards or technology used to design the software are within the scope of this document.

It is outside the scope of this document to describe what types are bugs are going to be involved,or how they will be solved, and how certain backend implementation may happen. It is also outside the scope of this document to describe in any detail at all how certain mentioned standards or technologies work and operate.

Definitions and Acronyms

Table of Definitions, Acronyms, and Abbreviations

Definition, Acronym, or Abbreviation	**Description**
SDS	Software Design Specification.

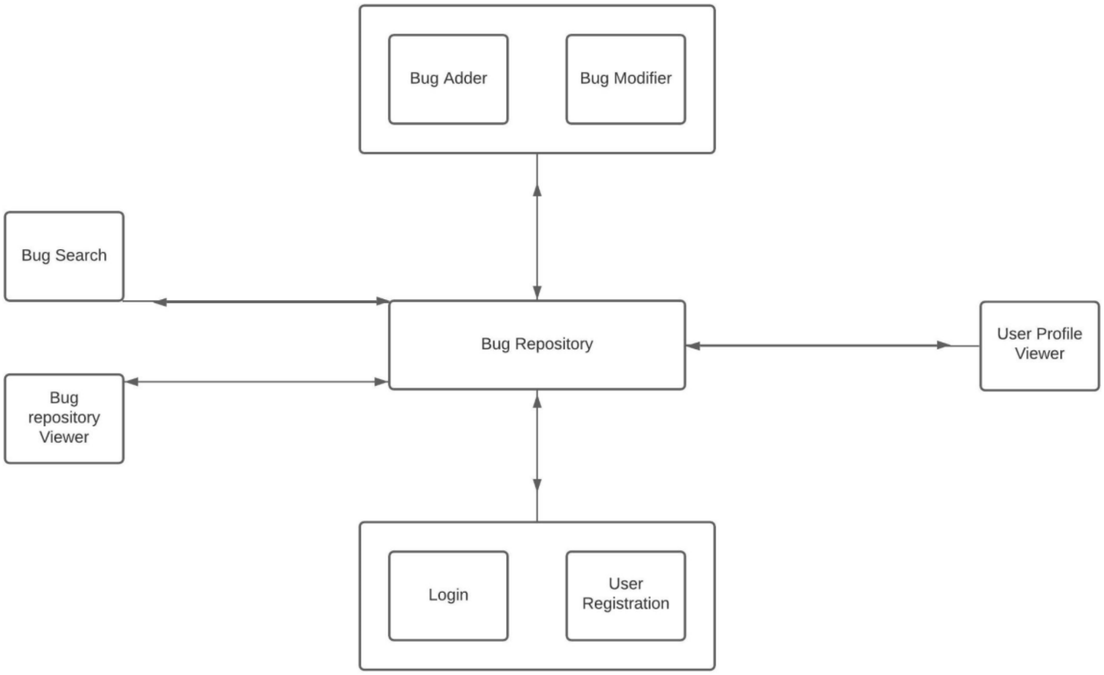
References

Table of References

References	**Description**
Software Development Plan	The Software Development Plan from the Electronic Stamp project was referenced.
Software Requirements Specification	The Software Requirements Specification from the Electronic Stamp project was referenced.

2. Decomposition Description

Module Decomposition



Repository Model

BugVITa has been decomposed into the following modules.

- Bug Repository : This module collects data from the user to be used for logging Bugs and issues for a particular Product(project) and also includes User data.
- User Profile : This module is responsible for displaying each User's profile details. It is also responsible for editing profile details like "password"
- Bug Add and Modify: Responsible to update the Bug repository by adding new Bug Reports or by modifying existing Bug reports.
- Bug search: Enables user to search the bug repository by providing search parameters relevant to a particular Bug report object.
- Bug repository viewer: Fetches the list of all bugs pertaining to a particular product from the repository.
- User Login and Registration:
  - Login : Facilitates user login by validating inputted user credentials against the corresponding credentials in the User Database
  - User registration: Enables new users to sign-up by creating their own password and adding their credentials to the User Database

Concurrent Process Decomposition

BugVITa Bug Management System is a simple tool that provides a list of errors and bugs encountered by the developers during the development cycle.
In particular, there are no such concurrent processes taking place at any point in the application. The behavior of the system is such that only one functionality can be used at a time.

Data Decomposition

The following are the two major data components, the User Information Table and all the Bugs List Tables.

User Information Table: This is a database that contains the following data items;

- User Primary key (User ID) : unique number assigned to each user.
- User Email-ID: A string-containing Email of the user. Used for logging into the system as the username and as a unique identifier for each us
- Password: A Password to obtain access to the application..
- Products associated: A list strings specifying the Products(Bug List Tables) associated with each user, that they can access.
- Privilege: Whether they are Tester,Developer,Admin etc.

Each Bugs-list Table: This is a database that stores a list of all the Bug-Reports that have the details of each Bug that has been logged.

- Bug ID: Unique ID for a bug that functions as a primary key.
- Bug Name/Title
- Bug Description
- Bug author
- Affects version
- Components affected
- Assignee
- Bug Date(Last modified)
- Priority
- Issue status

3. Dependency Description

Inter-module Dependencies

3.1.1. Independent Modules

The following modules are independent and do not rely on any other modules to initiate them or to provide data.

- Bug Page
- User

3.1.2. Dependent Modules

The following modules are dependent on one another for their functioning.

- Product: The Product module is a child of the Bug module and can not exist without a bug module where it uses it to create, store and modify bugs
- Bug List: This is a composition of a User module and every user module will have it's own bug list where it will have information about the bugs stored in it
- Developer Module: The Developer module is an extension of the User module which have a composition module of the bugs assigned to the developer
- Super User: This module is dependent on the user module and is a child of it.

Data Dependencies

The following Data Flow Diagram shows the data dependencies between the various entities and modules.

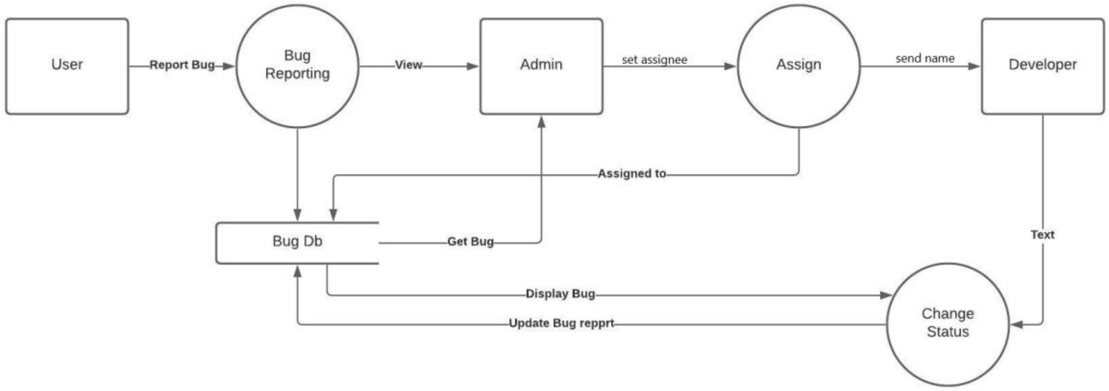


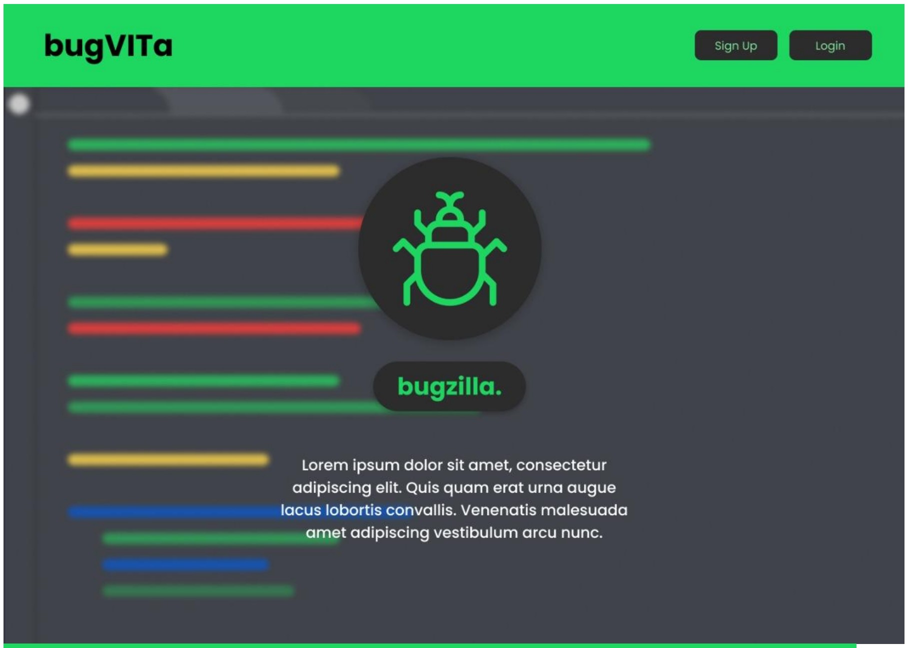
Figure 1, Data Flow Diagram

4. Interface Description

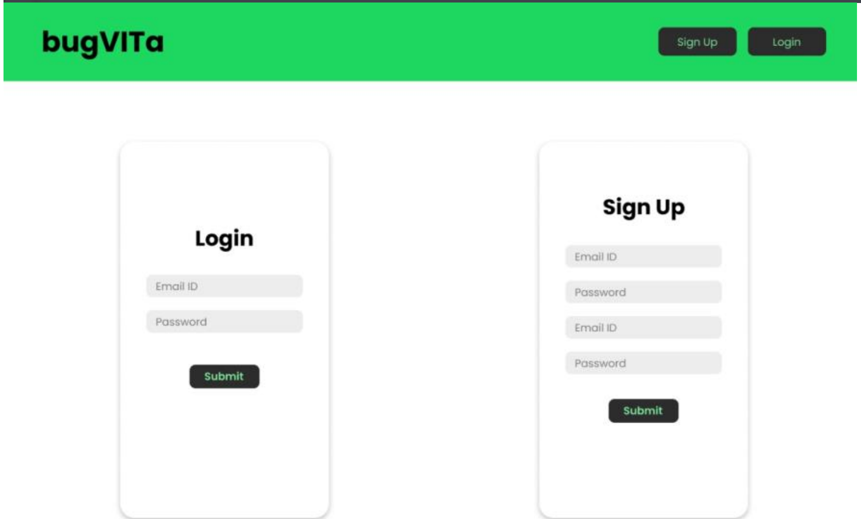
Module Interface

4.1.1. User Login/Signup Module Description

4.1.1.1. User Interface Design



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quis quam erat urna augue lacus lobortis convallis. Venenatis malesuada amet adipiscing vestibulum arcu nunc.

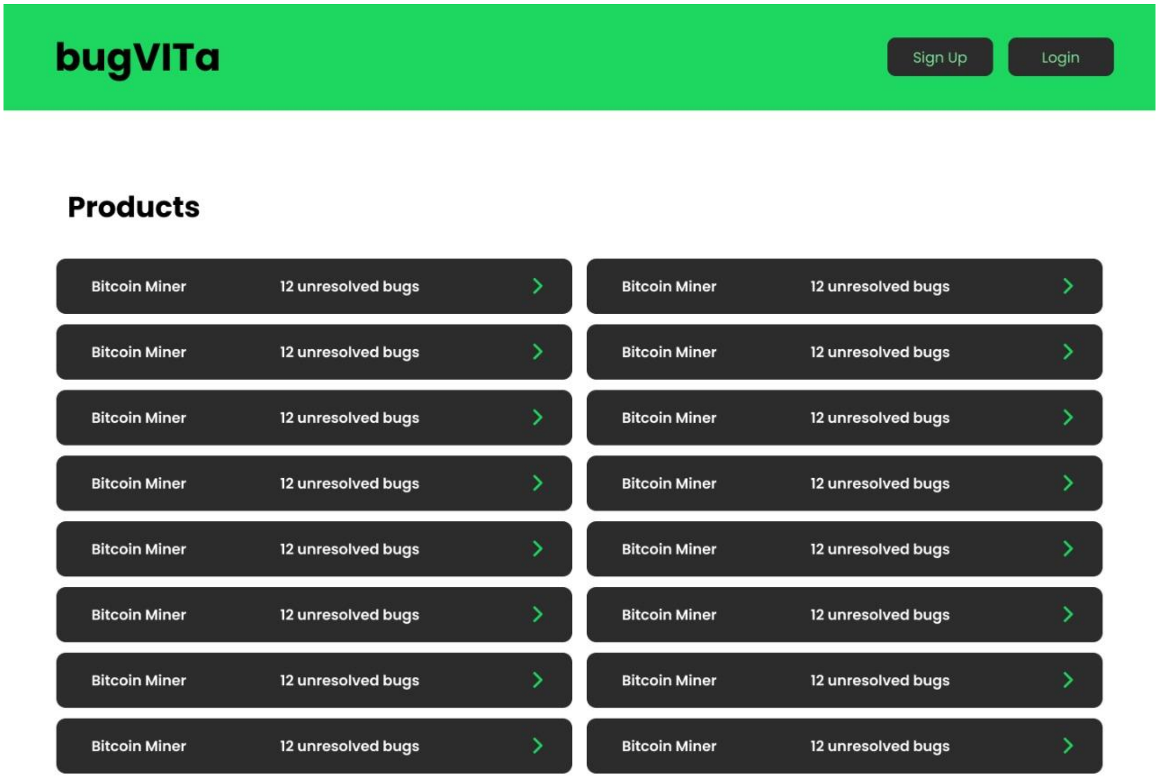


4.1.1.2. Description

Before the users can use the application, they must login first.
If the user has forgotten their password then they must contact the Database administrator to reset their password and provide a temporary password.
If the user is not registered in the User Database, they can Sign-up by entering their Email-ID and desired password and confirming the same.
At Sign-up, if the entered Email ID is already registered in the User Database, then the user will get a message saying so.

4.1.2. Product selection Module Description

4.1.2.1. User Interface Design



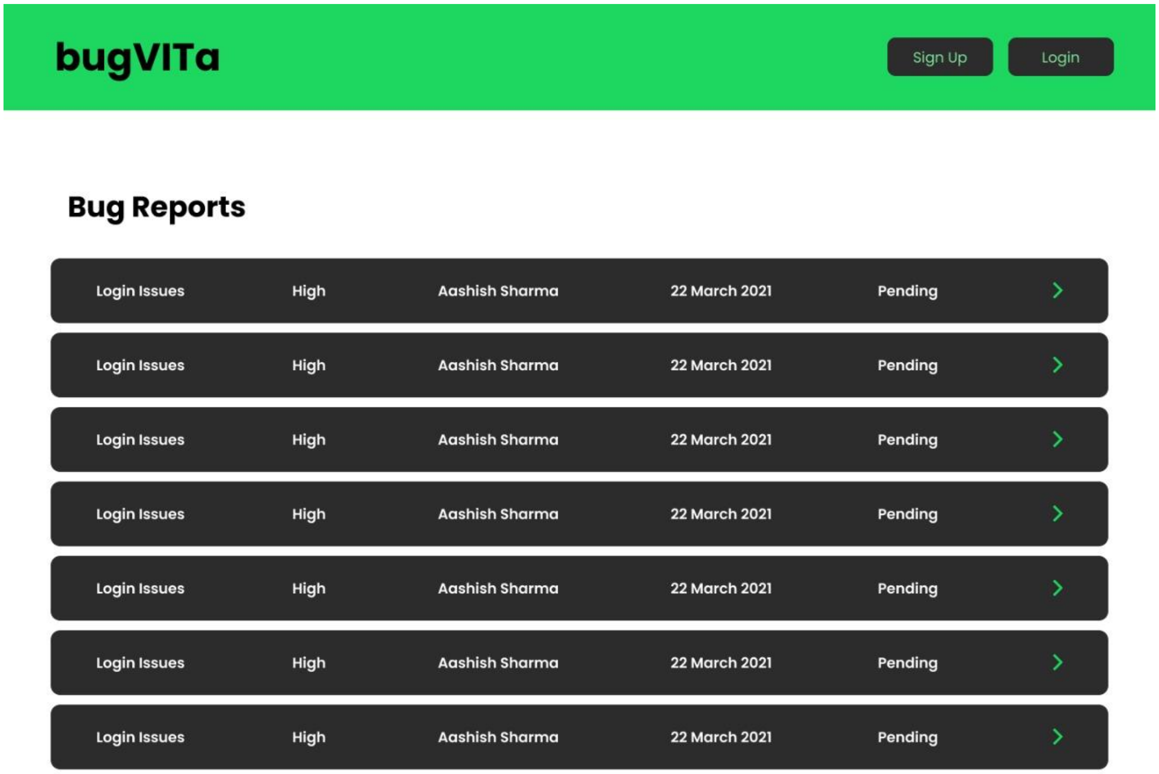
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs
Bitcoin Miner	12 unresolved bugs	Bitcoin Miner	12 unresolved bugs

4.1.2.2. Description

The following figures show the user interfaces for this module.
The user can select the Product (project) he is working on. This will bring the user to the Bug list page next, displaying all the bugs pertaining to that product.

4.1.3. Bugs List Module Description

4.1.3.1. User Interface Design



Bug Reports

Login Issues	High	Aashish Sharma	22 March 2021	Pending	>
Login Issues	High	Aashish Sharma	22 March 2021	Pending	>
Login Issues	High	Aashish Sharma	22 March 2021	Pending	>
Login Issues	High	Aashish Sharma	22 March 2021	Pending	>
Login Issues	High	Aashish Sharma	22 March 2021	Pending	>
Login Issues	High	Aashish Sharma	22 March 2021	Pending	>
Login Issues	High	Aashish Sharma	22 March 2021	Pending	>

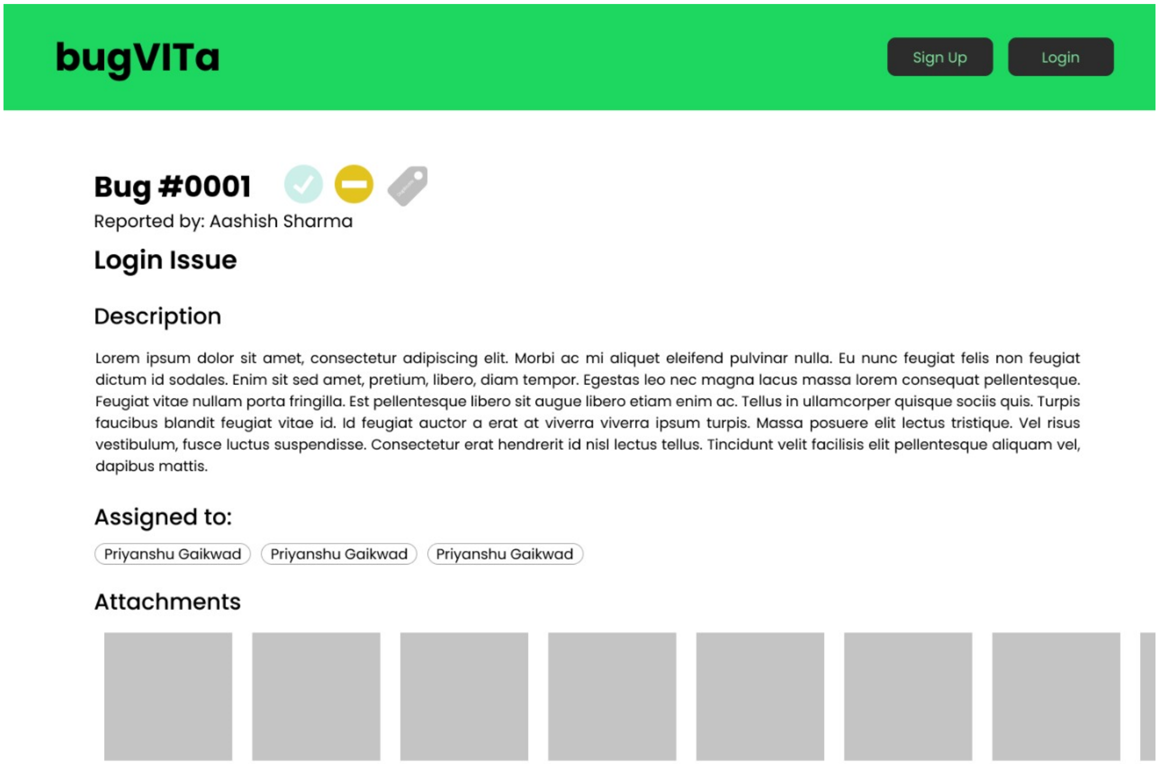
4.1.3.2. Description

This Interface allows the user to browse the list of all bugs pertaining to the selected product.
The user can click on a particular bug to view it.
Users can also sort the list according to each attribute(e.g:alphabetical order of names,highest priority first, etc.)

The user will also have a button here “Add Bug”, to add a bug related to the product.

4.1.4. Bug report Module Description

4.1.4.1. User Interface Design



Bug #0001

Reported by: Aashish Sharma

Login Issue

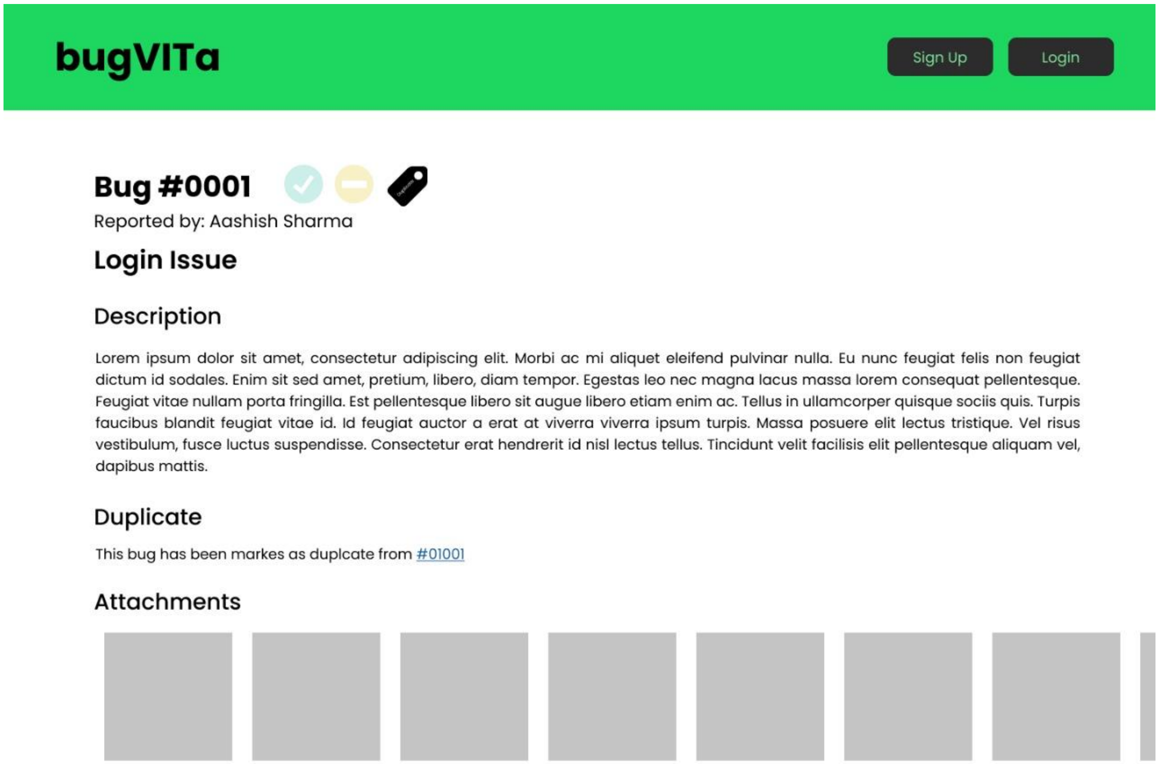
Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac mi aliquet eleifend pulvinar nulla. Eu nunc feugiat felis non feugiat dictum id sodales. Enim sit sed amet, pretium, libero, diam tempor. Egestas leo nec magna lacus massa lorem consequat pellentesque. Feugiat vitae nullam porta fringilla. Est pellentesque libero sit augue libero etiam enim ac. Tellus in ullamcorper quisque sociis quis. Turpis faucibus blandit feugiat vitae id. Id feugiat auctor a erat at viverra viverra ipsum turpis. Massa posuere elit lectus tristique. Vel risus vestibulum, fusce luctus suspendisse. Consectetur erat hendrerit id nisl lectus tellus. Tincidunt velit facilisis elit pellentesque aliquam vel, dapibus mattis.

Assigned to:

Priyanshu Gaikwad Priyanshu Gaikwad Priyanshu Gaikwad

Attachments



Bug #0001

Reported by: Aashish Sharma

Login Issue

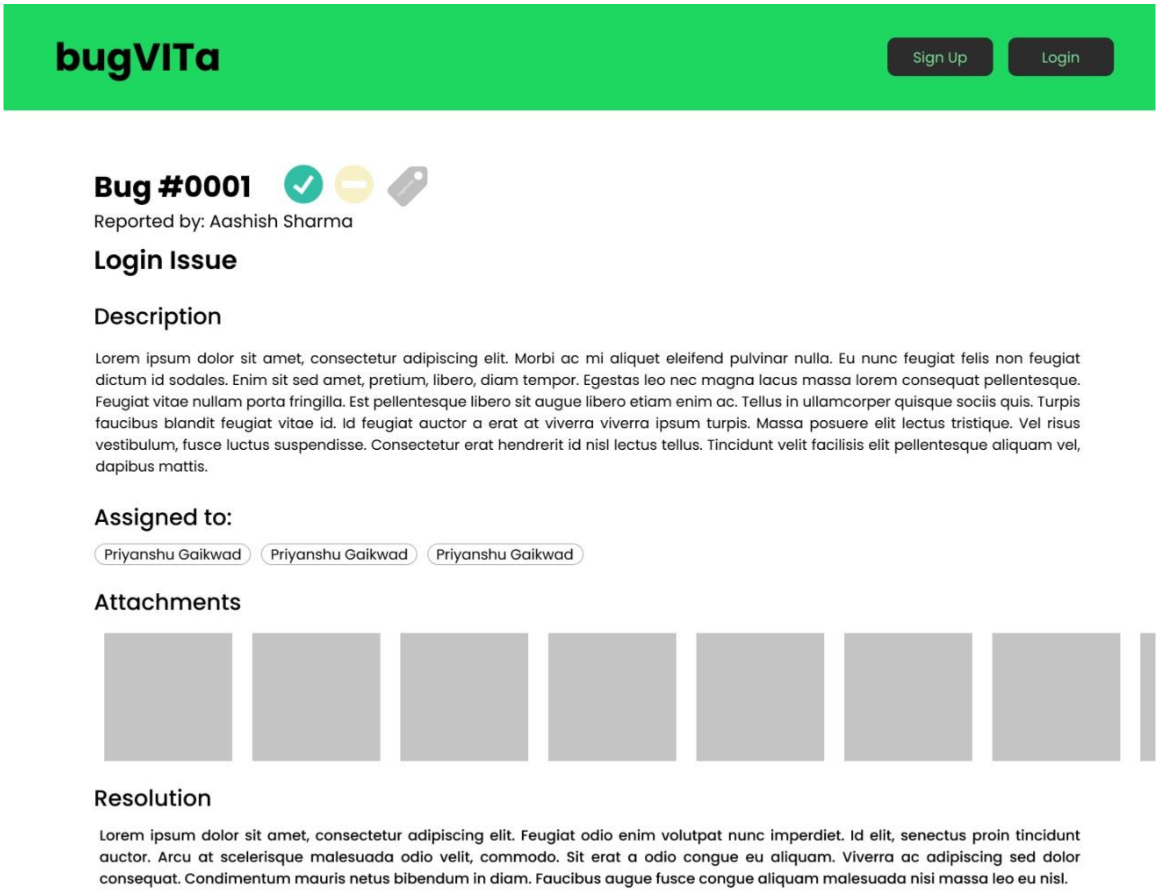
Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac mi aliquet eleifend pulvinar nulla. Eu nunc feugiat felis non feugiat dictum id sodales. Enim sit sed amet, pretium, libero, diam tempor. Egestas leo nec magna lacus massa lorem consequat pellentesque. Feugiat vitae nullam porta fringilla. Est pellentesque libero sit augue libero etiam enim ac. Tellus in ullamcorper quisque sociis quis. Turpis faucibus blandit feugiat vitae id. Id feugiat auctor a erat at viverra viverra ipsum turpis. Massa posuere elit lectus tristique. Vel risus vestibulum, fusce luctus suspendisse. Consectetur erat hendrerit id nisl lectus tellus. Tincidunt velit facilisis elit pellentesque aliquam vel, dapibus mattis.

Duplicate

This bug has been markes as duplcate from #01001

Attachments



Bug #0001

Reported by: Aashish Sharma

Login Issue

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac mi aliquet eleifend pulvinar nulla. Eu nunc feugiat felis non feugiat dictum id sodales. Enim sit sed amet, pretium, libero, diam tempor. Egestas leo nec magna lacus massa lorem consequat pellentesque. Feugiat vitae nullam porta fringilla. Est pellentesque libero sit augue libero etiam enim ac. Tellus in ullamcorper quisque sociis quis. Turpis faucibus blandit feugiat vitae id. Id feugiat auctor a erat at viverra viverra ipsum turpis. Massa posuere elit lectus tristique. Vel risus vestibulum, fusce luctus suspendisse. Consectetur erat hendrerit id nisl lectus tellus. Tincidunt velit facilisis elit pellentesque aliquam vel, dapibus mattis.

Assigned to:

Priyanshu Gaikwad Priyanshu Gaikwad Priyanshu Gaikwad

Attachments

Resolution

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Feugiat odio enim volutpat nunc imperdiet. Id elit, senectus proin tincidunt auctor. Arcu at scelerisque malesuada odio velit, commodo. Sit erat a odio congue eu aliquam. Viverra ac adipiscing sed dolor consequat. Condimentum mauris netus bibendum in diam. Faucibus augue fusce congue aliquam malesuada nisi massa leo eu nisl.

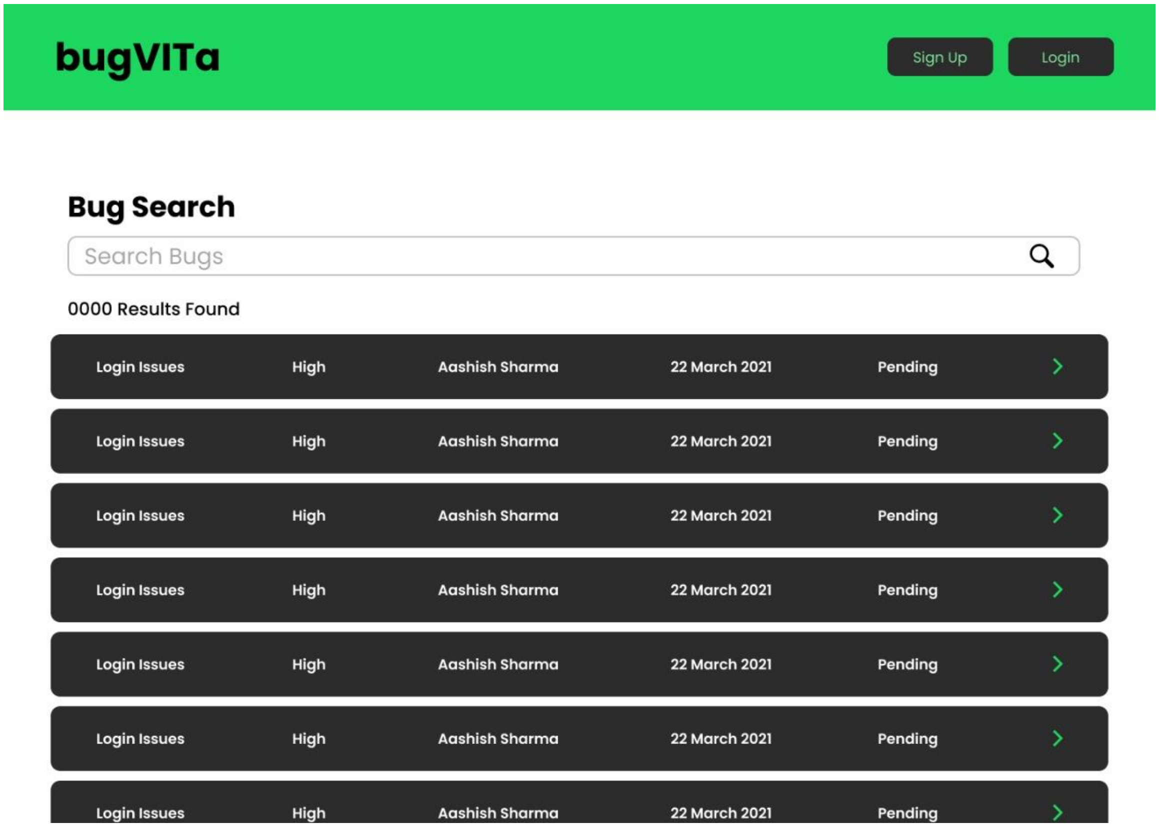
4.1.4.2. Description

Whenever a user clicks on a Bug object from either the Search results page or the Bug list page, they get the Bug Report/Summary in this interface.

The User will also have a button to modify the bug(if permitted), which will take them to the bug edit page.

4.1.5. Bug Search Module Description

4.1.5.1. User Interface Design



Bug Search

0000 Results Found

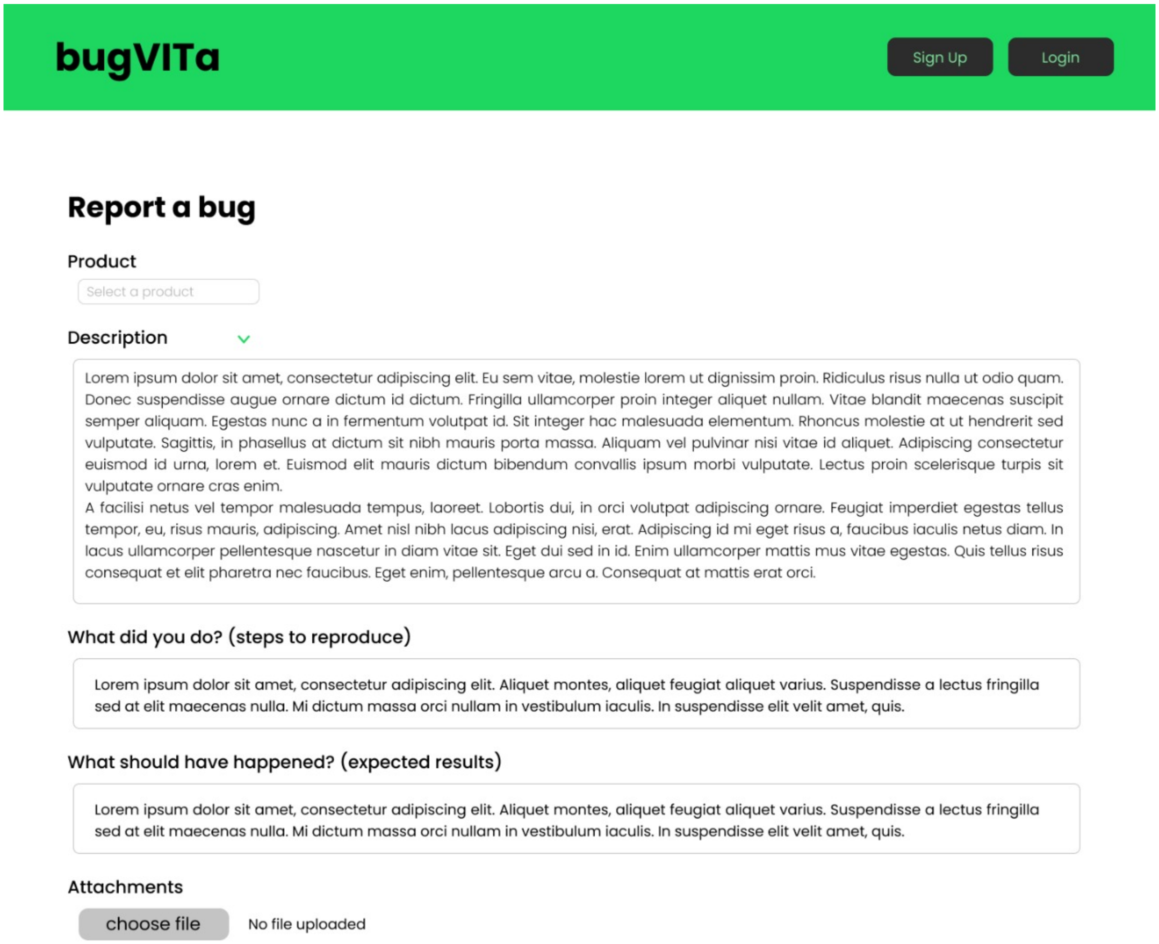
Login Issues	High	Aashish Sharma	22 March 2021	Pending
Login Issues	High	Aashish Sharma	22 March 2021	Pending
Login Issues	High	Aashish Sharma	22 March 2021	Pending
Login Issues	High	Aashish Sharma	22 March 2021	Pending
Login Issues	High	Aashish Sharma	22 March 2021	Pending
Login Issues	High	Aashish Sharma	22 March 2021	Pending
Login Issues	High	Aashish Sharma	22 March 2021	Pending

4.1.5.2. Description

Users can search for a Bug, and will receive relevant search results as shown in the interface. The user can click on any Bug object to view it’s detailed Bug Report.

4.1.6. Bug edit/add Module Description

4.1.6.1. User Interface Design



Report a bug

Product

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eu sem vitae, molestie lorem ut dignissim proin. Ridiculus risus nulla ut odio quam. Donec suspendisse augue ornare dictum id dictum. Fringilla ullamcorper proin integer aliquet nullam. Vitae blandit maecenas suscipit semper aliquam. Egestas nunc a in fermentum volutpat id. Sit integer hac malesuada elementum. Rhoncus molestie at ut hendrerit sed vulputate. Sagittis, in phasellus at dictum sit nibh mauris porta massa. Aliquam vel pulvinar nisi vitae id aliquet. Adipiscing consectetur euismod id urna, lorem et. Euismod elit mauris dictum bibendum convallis ipsum morbi vulputate. Lectus proin scelerisque turpis sit vulputate ornare cras enim.
A facilisi netus vel tempor malesuada tempus, laoreet. Lobortis dui, in orci volutpat adipiscing ornare. Feugiat imperdiet egestas tellus tempor, eu, risus mauris, adipiscing. Amet nisl nibh lacus adipiscing nisi, erat. Adipiscing id mi eget risus a, faucibus iaculis netus diam. In lacus ullamcorper pellentesque nascetur in diam vitae sit. Eget dui sed in id. Enim ullamcorper mattis mus vitae egestas. Quis tellus risus consequat et elit pharetra nec faucibus. Eget enim, pellentesque arcu a. Consequat at mattis erat orci.

What did you do? (steps to reproduce)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquet montes, aliquet feugiat aliquet varius. Suspendisse a lectus fringilla sed at elit maecenas nulla. Mi dictum massa orci nullam in vestibulum iaculis. In suspendisse elit velit amet, quis.

What should have happened? (expected results)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquet montes, aliquet feugiat aliquet varius. Suspendisse a lectus fringilla sed at elit maecenas nulla. Mi dictum massa orci nullam in vestibulum iaculis. In suspendisse elit velit amet, quis.

Attachments

4.1.6.2. Description

When the user clicks on “Add Bug” button from another page OR when the user clicks on “Modify Bug” button from another page, they are brought to this page to edit the Bug Report.
Once finished, they can click “Save” to add/update the Bugs Database Table.

Process Interface

4.1.7. Login

The primary objective of this module is to provide the user with a log in and a sign up interface so that when a user reports a bug he is already registered in the database thus able to access the methods to call and create bugs

The module will require an Email/username and a password for log in. To register the user will have to provide Email, Password, Retype the Password and Username.

4.1.8. Bug List

The user can look at a list of bugs that are added by other users and look at them and their resolutions.The bug list provides the user with a list of bugs and some information about who added the bug and also an option to further open the bug report for further querries of the user.

4.1.9. Bug Report Description

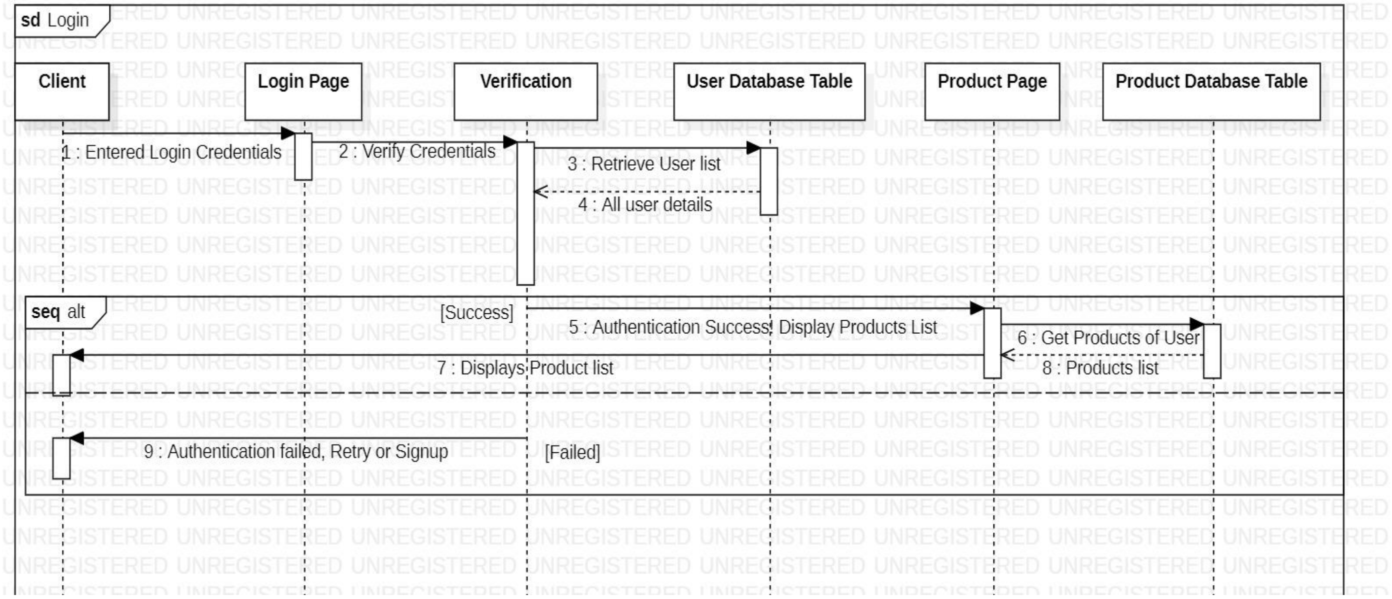
This module will have all the description given by the reporter to the system, this Interface will display the description of the bug also how the user who reported it intended the feature to work.The page will have the current status of the bug resolution whether it has been resolved or not or is marked as a duplicate report which has been resolved already. The bug is unresolved will also have the names of the developers assigned that are currently working on the bug fix.The Interface will also have attachments if the user provides any screenshots of the bug encountered.

4.1.10. Bug Search Module Description

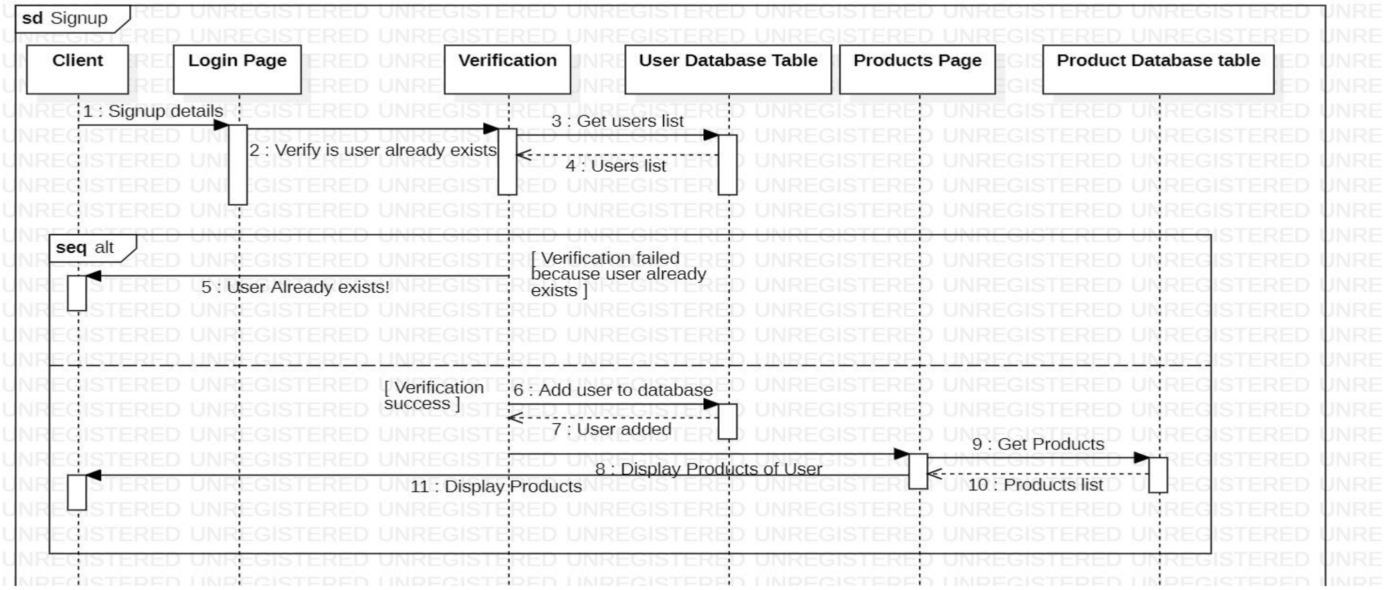
This Interface is responsible to display bugs specific to the user's input and query the page has a text input where the user can input a string related to the bug report of the keyword he wants to see, on clicking the search button the backend returns a list of possible matches to the query by fetching the bugs that has the keyword in their name or description in the database of the bugs.

5.1 Sequence Diagrams (Using StarUML)

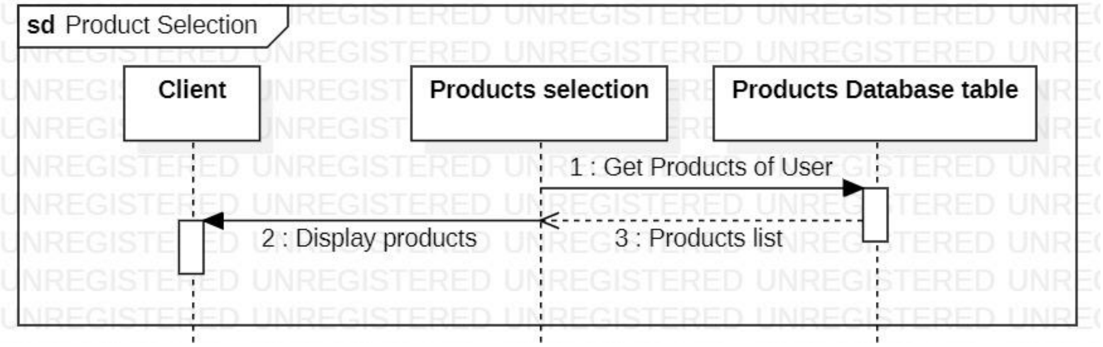
Login Sequence



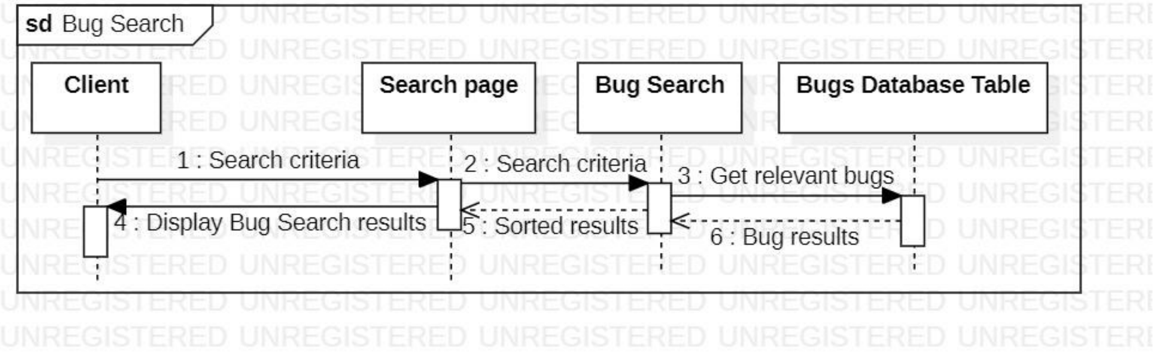
Signup sequence



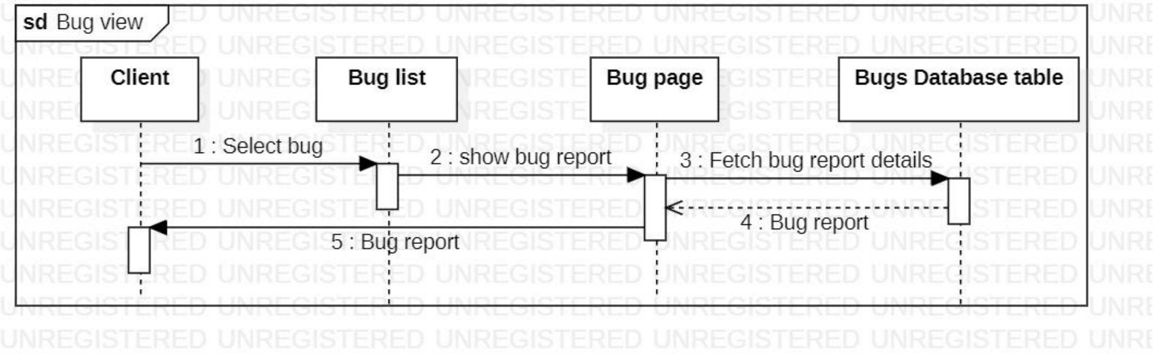
Product selection sequence



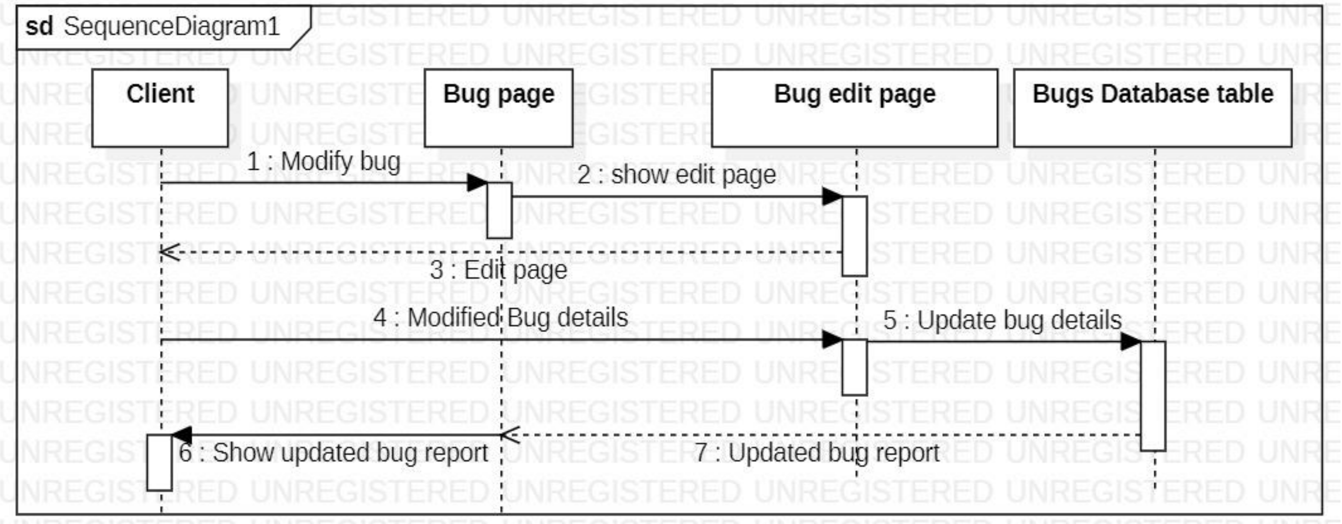
Bug search sequence



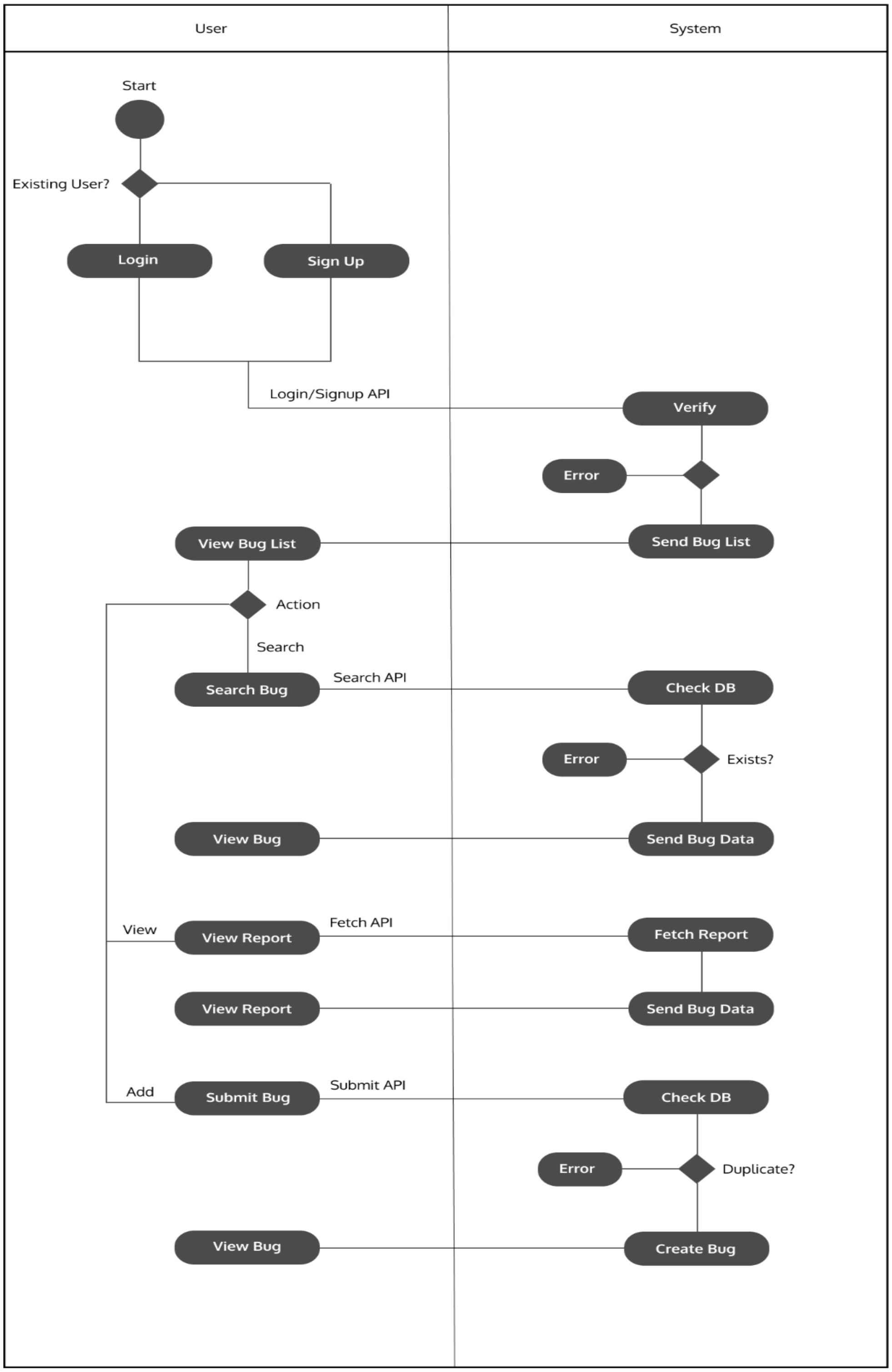
Bug View sequence



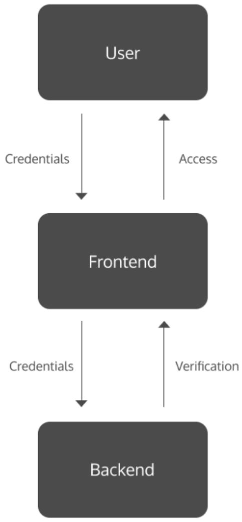
Bug Add/Modify sequence

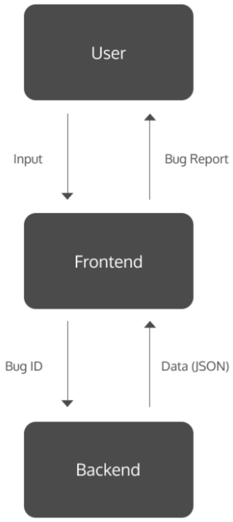


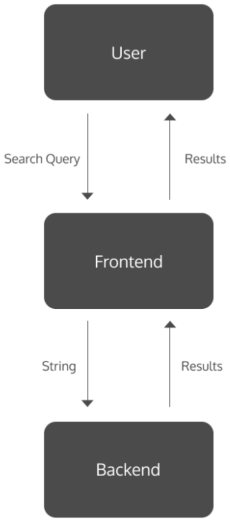
5.2 Activity Swimlane Diagram

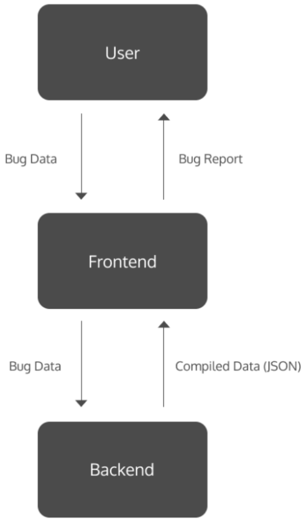


5.3 Collaboration Diagram

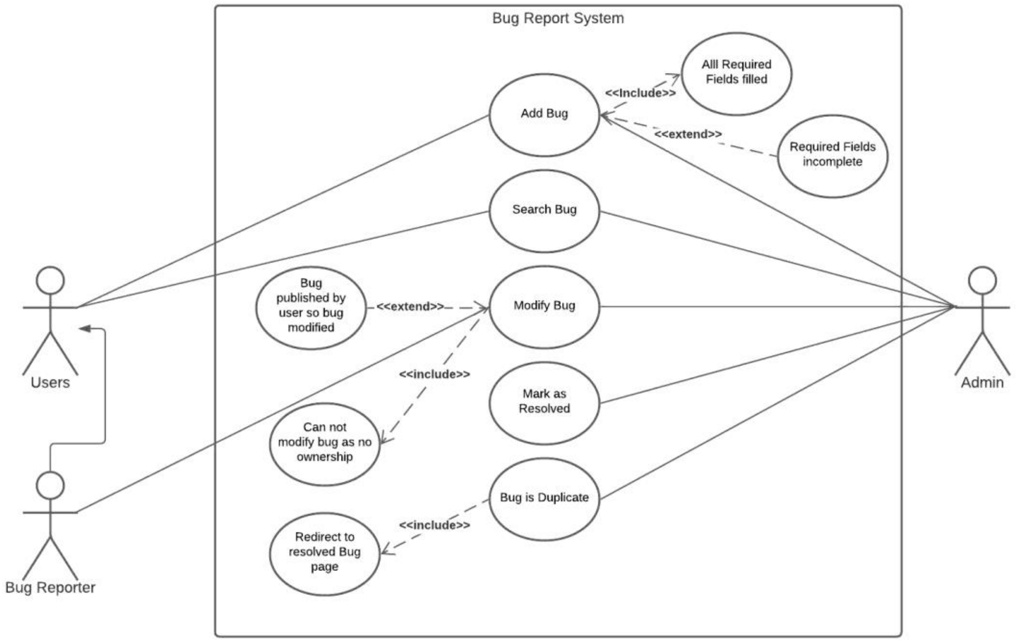

Login/ Signup


View Report


Search

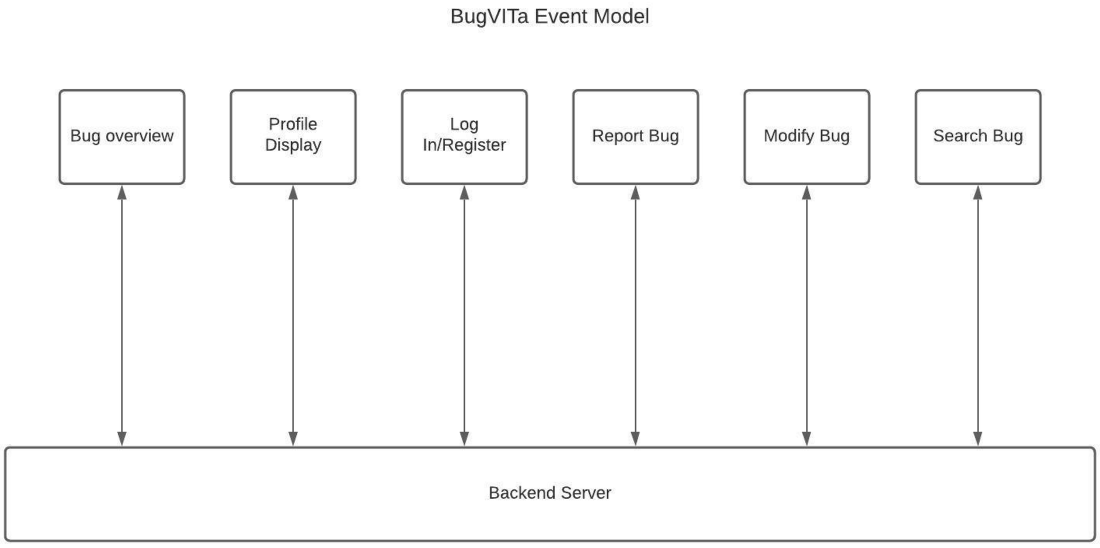

Submit Report

5.4 Use Case Diagram


Bug Report System
Add Bug
<<Include>>
Alll Required Fields filled
<<extend>>
Required Fields incomplete
Search Bug
Bug published by user so bug modified
<<extend>>
Modify Bug
<<include>>
Can not modify bug as no ownership
Mark as Resolved
Bug is Duplicate
<<include>>
Redirect to resolved Bug page
Users
Bug Reporter
Admin

Use Case diagram

5.5 Control System



Broadcast Model

6 Detailed Design

Module Detailed Design

6.1.1 User Module

6.1.1.1 Design

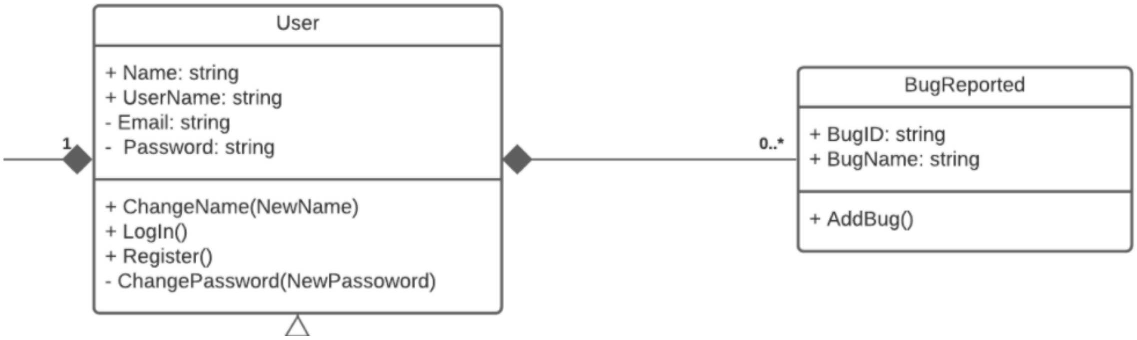


Figure 14, User and Bug Design

6.1.1.2 Design Description

The user class is the main class that instantiates and calls the other classes to create new processes like creating new bug report or searching or signing in.It has 4 items Name, UserName, Email, Password. The fields Email and Password are made private so to keep the data protected.

The User class has a component to it called the BugReport, Where the ID and name of the bugs reported by the user are stored. So that the user can find his bugs easily and modify them if he wished to do so

The User has 4 methods to it namely ChangeName, LogIn, Register and ChangePassword. These methods are used to modify the user’s public and private information and make user login into their account and register for one if they don’t have one.

6.1.2 User Hierarchy Design

6.1.2.1 Design

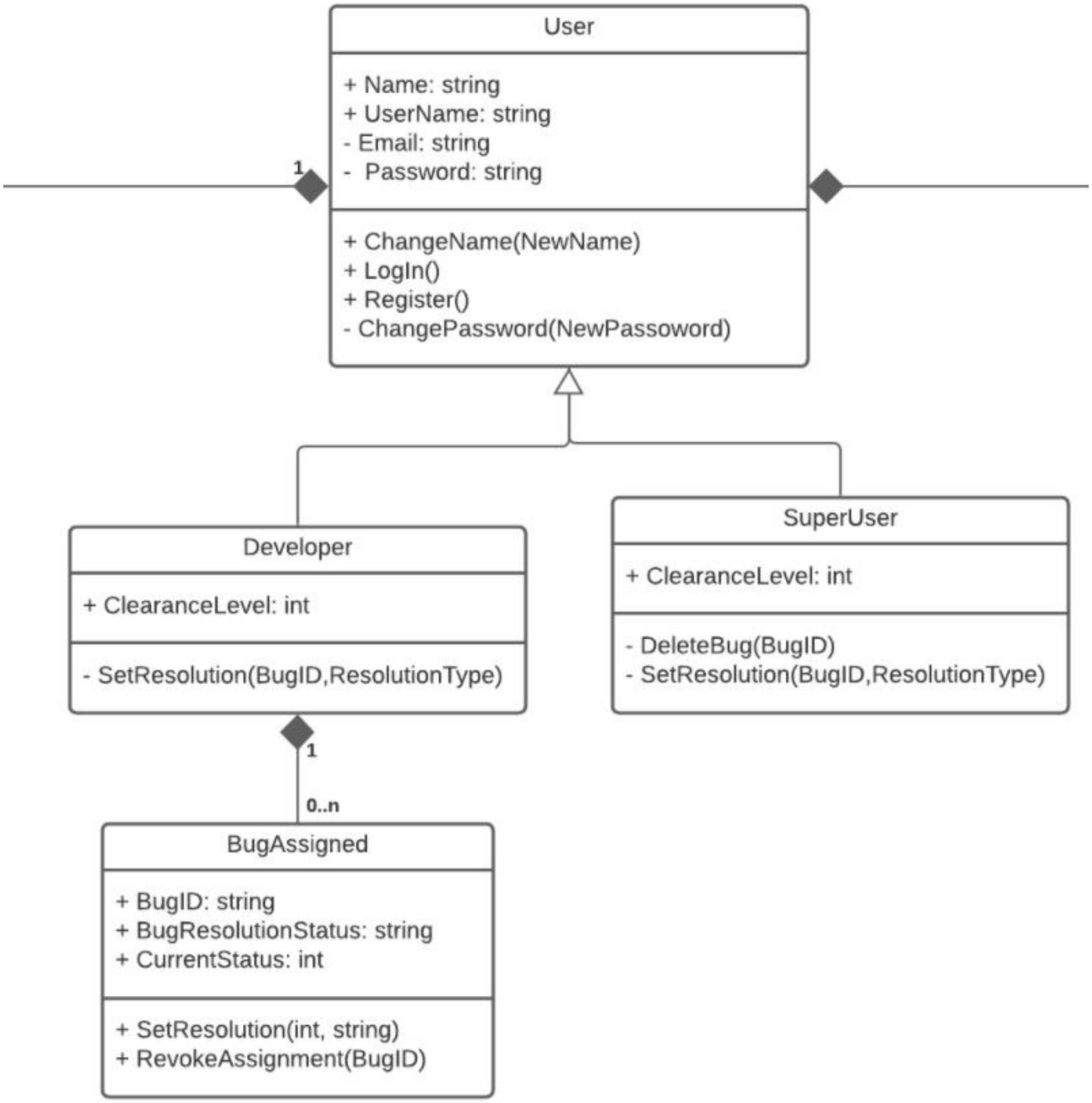


Figure 15, User Hiararchy

6.1.2.2 Design Description

The are also some users that are given special privileges or are different from a general user. There are 3 types of users namely the User, the SuperUser and the Developer.

The User has default and the least amount of privileges and can report and modify their own bugs. The Developer is a User with an extra privilege such as a component of BugAssigned, this will have a list of bugs he is assigned to. He can also Revoke the assignment of the bug assigned to him and also set a resolution once the bug is resolved.

The SuperUser is a user with many privileges. He can remove any bugs, assign a bug to a developer and also remove users that have lower privilege level than them.

6.1.3 Bug Module Detailed Design

6.1.3.1 Design

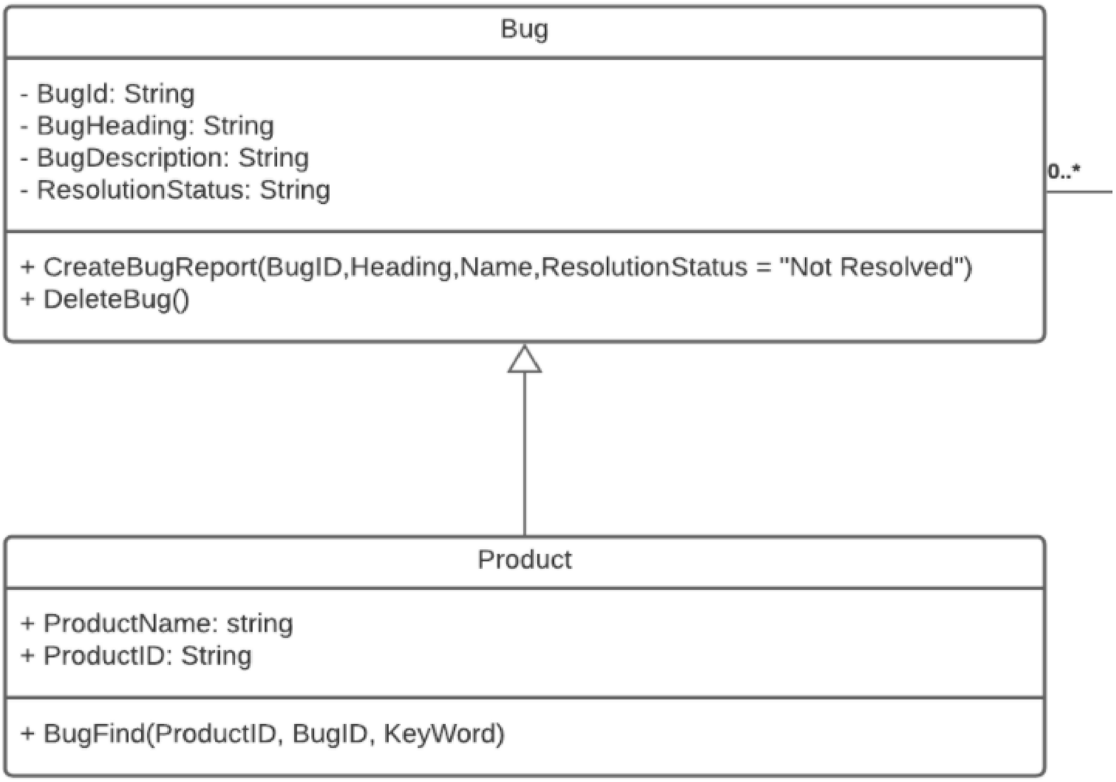

Bug
- BugId: String
- BugHeading: String
- BugDescription: String
- ResolutionStatus: String
+ CreateBugReport(BugID,Heading,Name,ResolutionStatus = "Not Resolved")
+ DeleteBug()
0..*
Product
+ ProductName: string
+ ProductID: String
+ BugFind(ProductID, BugID, KeyWord)

Figure 16, BugModule

6.1.3.2 Design Description

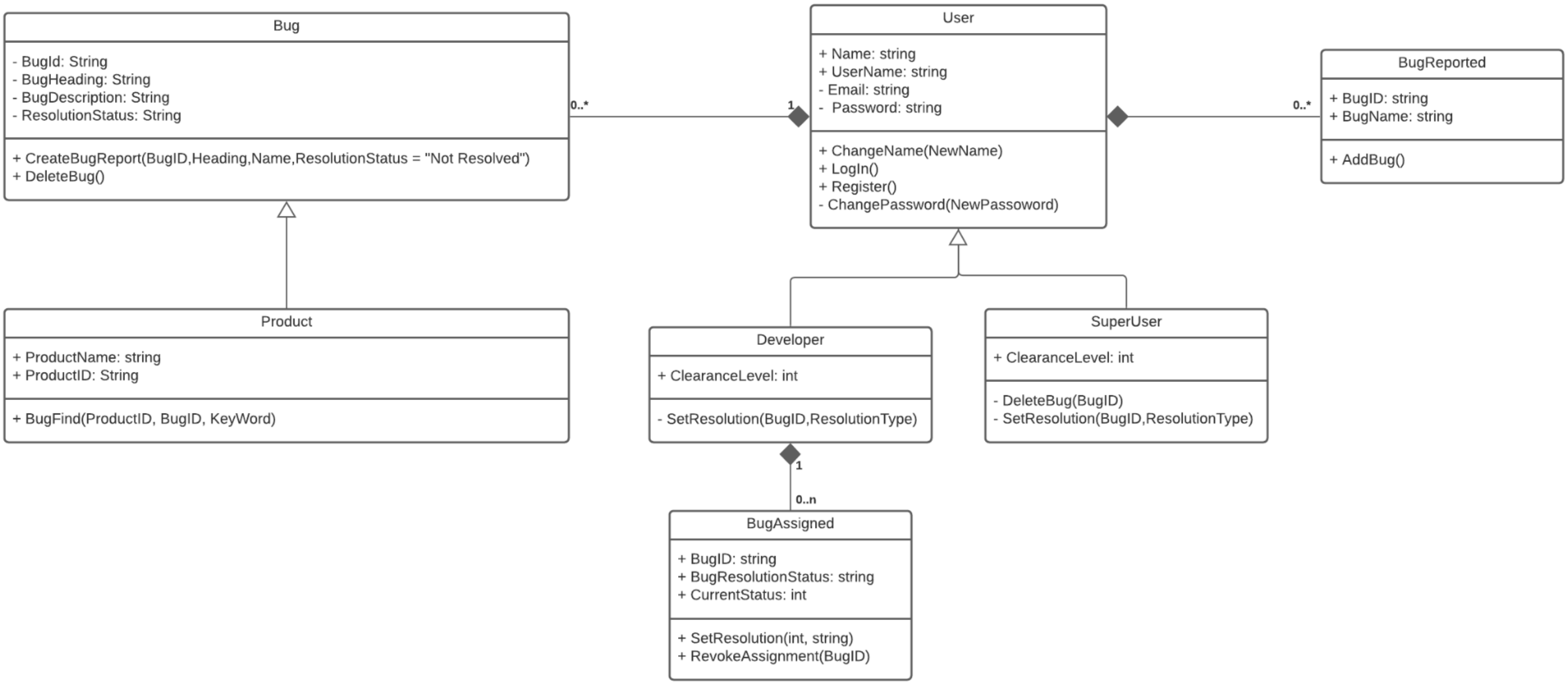
The Bug Class is a child of Product. The the bug section is organised in the following way

- The database has a product database
- The product key will have a database of the specified bugs

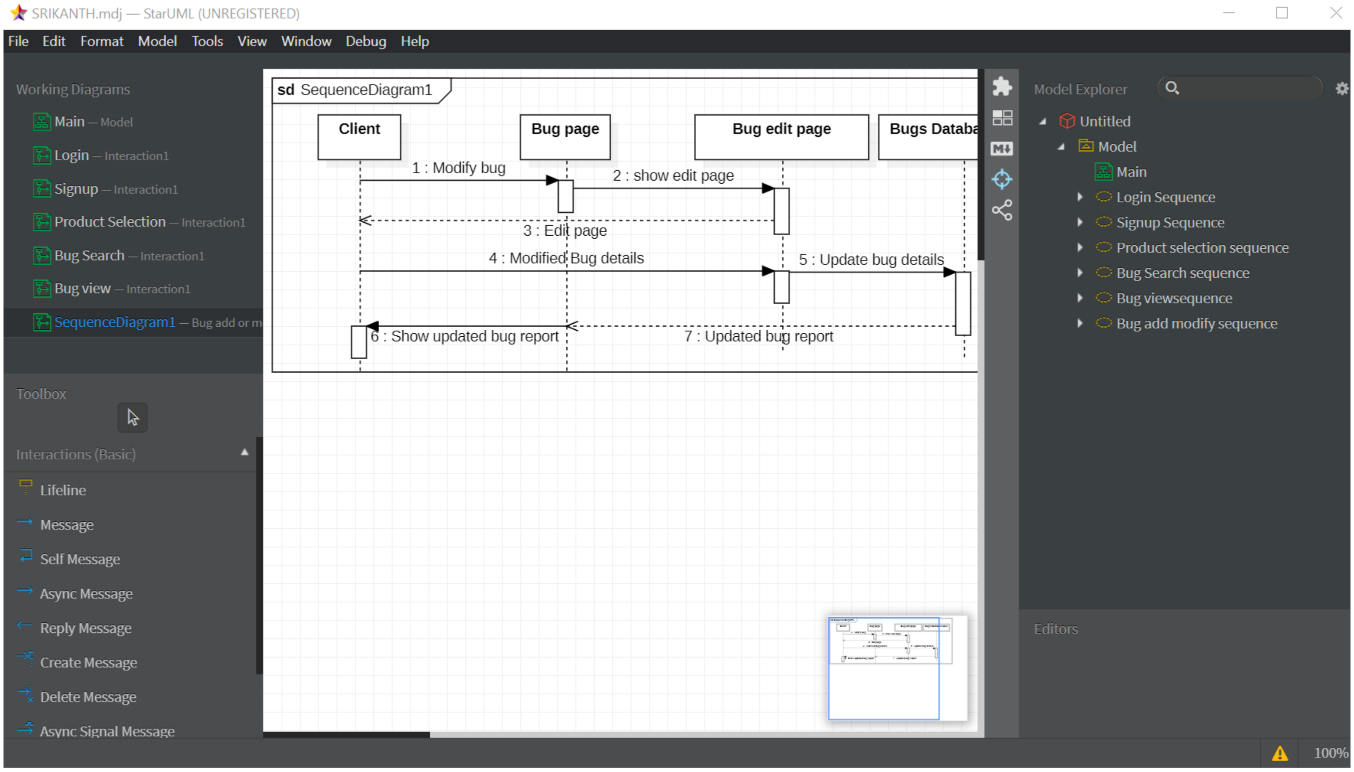
The product class will have access to only it’s bugs where it can instantiate and call the methods of it’s parent class.

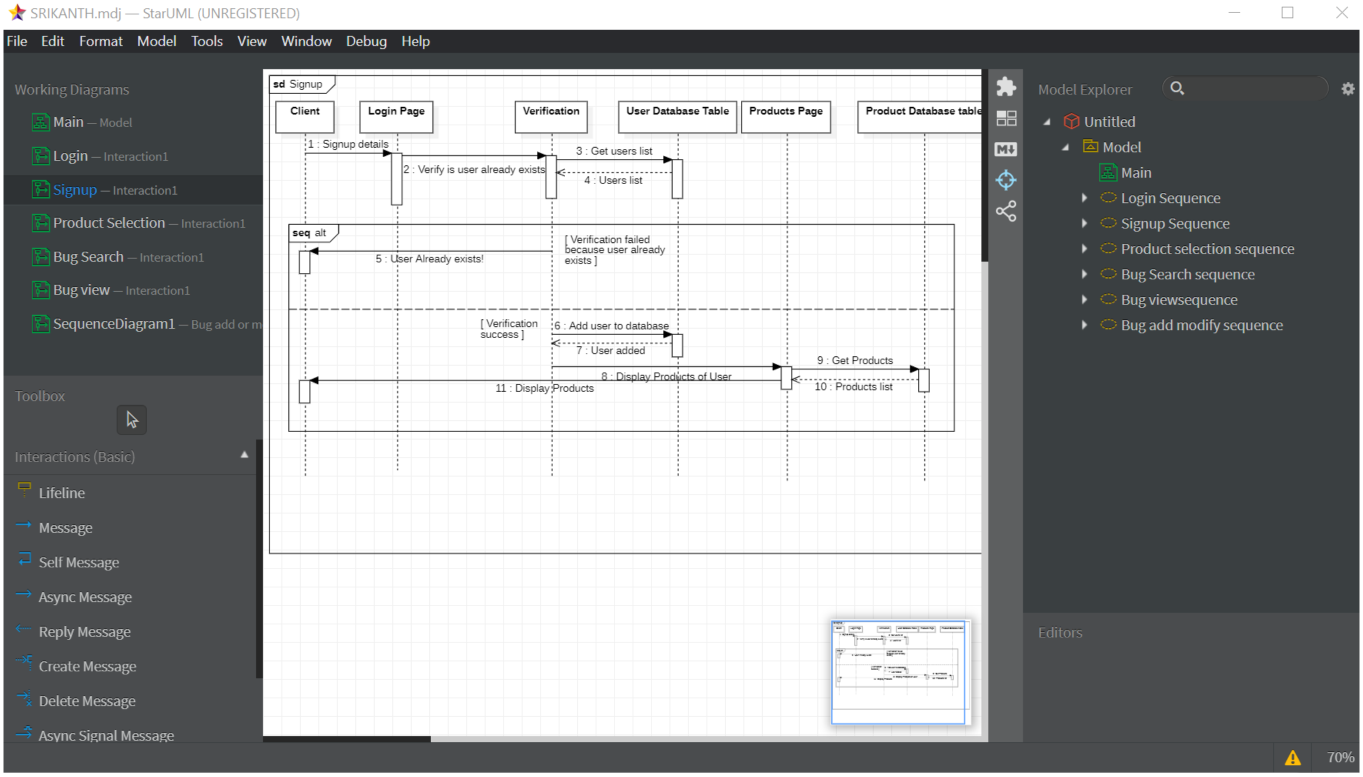
The User will have to create a bug by specifying the product of whom the bug is of, so the User will instantiate the bug class using the product class .

Appendix A – BugVITa Class Diagram

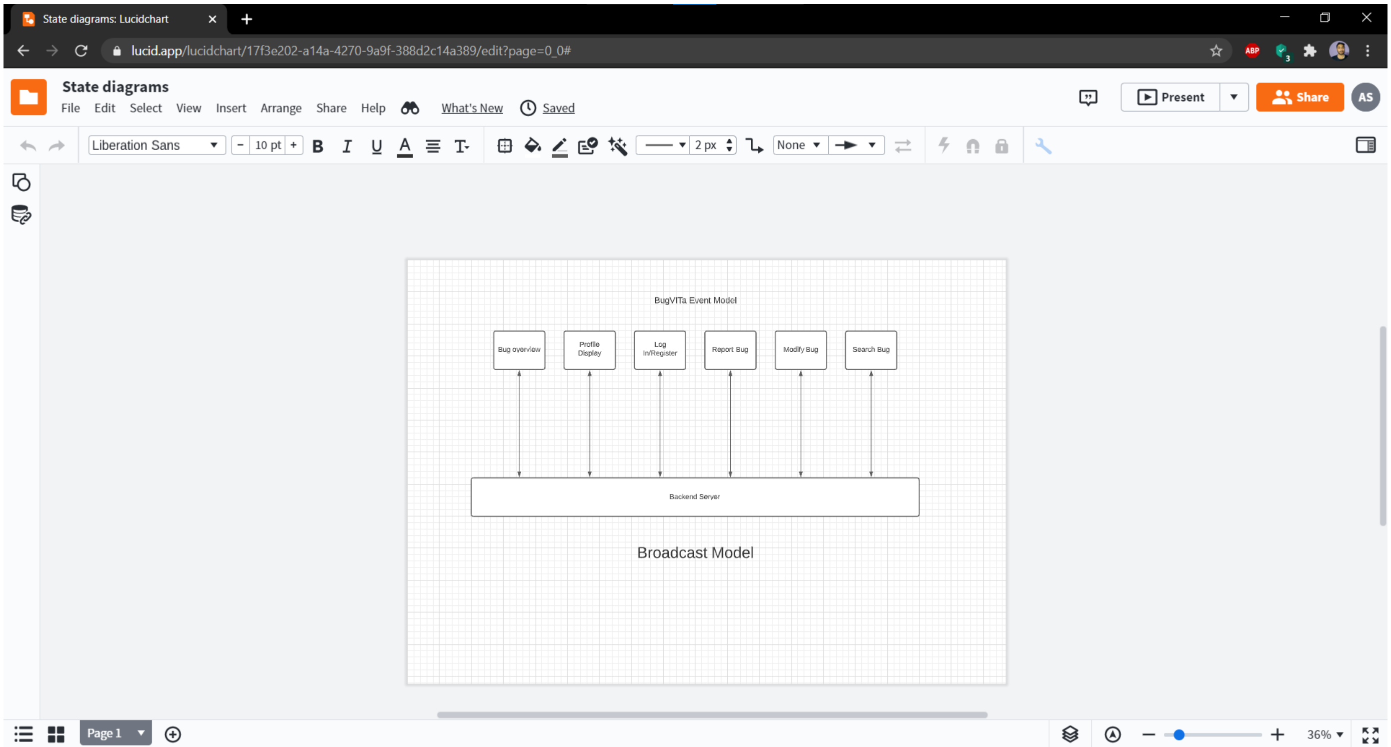


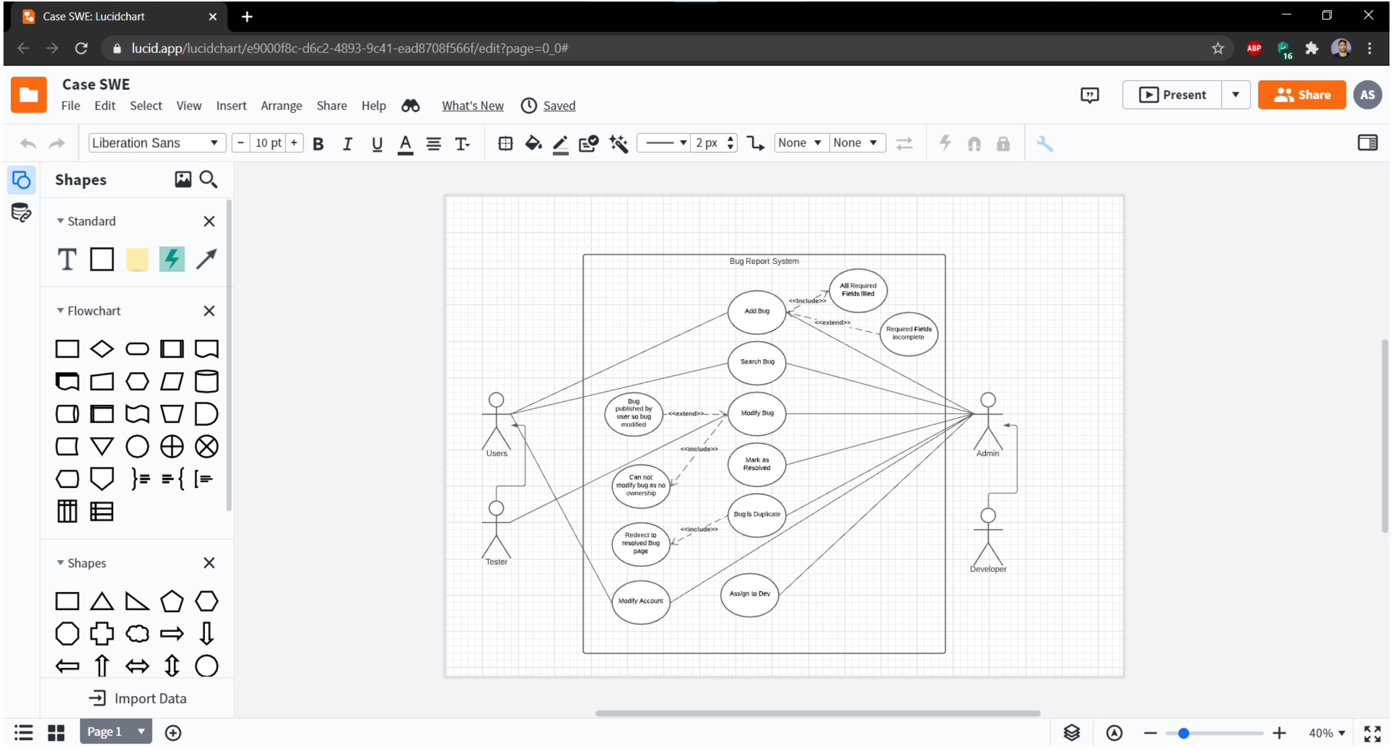
Appendix B - Sequence diagram sample screenshots[StarUML]





Appendix C - Case and State screenshots[LucidChart]


State diagrams: Lucidchart
lucid.app/lucidchart/17f3e202-a14a-4270-9a9f-388d2c14a389/edit?page=0_0#
State diagrams
File Edit Select View Insert Arrange Share Help What's New Saved
Present
Share
AS
Liberation Sans
10 pt
2 px
None
BugVITa Event Model
Bug overview
Profile Display
Log In/Register
Report Bug
Modify Bug
Search Bug
Backend Server
Broadcast Model
Page 1
36%


Case SWE: Lucidchart
lucid.app/lucidchart/e9000f8c-d6c2-4893-9c41-ead8708f566f/edit?page=0_0#
Case SWE
File Edit Select View Insert Arrange Share Help What's New Saved
Present
Share
AS
Liberation Sans
10 pt
2 px
None
None
Shapes
Standard
Flowchart
Shapes
Import Data
Bug Report System
Add Bug
<<Include>>
All Required Fields filled
<<extend>>
Required Fields incomplete
Search Bug
Bug published by user so bug modified
<<extend>>
Modify Bug
<<include>>
Can not modify bug as no ownership
Mark as Resolved
Bug Is Duplicate
<<include>>
Redirect to resolved Bug page
Modify Account
Assign to Dev
Users
Tester
Admin
Developer
Page 1
40%