P1-Pothole Tracking and Repair

Prasant Kumar (2018MB12036)
Kulsharest Jain (2018MB12006)
Navneet Thillaisthanam (2018MB12030)
Nishant Kumar (2018MB12016)

INDEX

Revision Control

Revision No.	Reason for Change	Revision Date	Changes Done
V1.0	First Draft Of Project	20/10/2018	UC Identification, and 2 UC Fully Dressed
V2.0	Iteration 2	16/11/2018	• System Operation Contracts • Interaction diagrams between objects - Sequence • Domain Class Design • User Interface Prototype - sketches - hand or with a tool
V3.0	Iteration 3	27/12/2018	• Updates from feedback review (see appendix 1) • Login page designed and added diagrams • Added use-case UC03-06; terse and fully-dressed format • Added Activity & Package Diagram • State chart Diagram update.
V4.0	Iteration 4	25/01/2019	
V5.0	Iteration 5	13/02/2018	

Reference:

UML Book: Applying UML and Patterns 3rd edition
UML Tool: StarUML
User Interface Prototype tool: mockplus

1. Project: P1 - Pothole Tracking and Repair

1.1. Requirement:

Bangalore Mayor has promised to repair all the potholes in the city with citizen participation. You are IT implementation partner to BBMP for web based or smartphone app based (or both) Pothole Tracking and Repair system. Citizens will report the potholes with a GPS Map location and a photograph. BBMP will assign pothole repair work to a Contractor. Contractor will provide a cost estimate along with a photograph and size confirmation of the pothole. The software will calculate the pothole size from photograph and cross check with the details submitted by Contractor. BBMP will approve the cost estimate and Contractor will perform the work. Citizen who have reported pothole will be able to report their satisfaction on repair work. Contracts will be assigned by area - constituency / wards / pin code. Contractor will be paid on a monthly basis. A Google Map integrated navigation of potholes with their status will be provided. The Mayor should be able to visualize the progress for overall city or by an area - constituency / pin code or by individual roads.

2. Identify Actors and Use Cases

2.1. UC01 - Login and Authentication:

Actors: Citizen, BBMP, Contractor
Use Case: System shall have login/logout feature for all users and for accessing the system, a user has to input login credentials, comprising of a UserID and password. While creating profile of Citizen and Contractor, the system shall have personal details of the user. BBMP shall have admin rights to make changes in data stored in the system.

2.2. UC02 - Report Pothole

Actors: Citizen
Use Case: The Citizens report the potholes with a GPS Map location and a photograph of the pothole. A Google Map will be integrated for reporting pothole

2.3. UC03 – Assign Pothole

Actors: BBMP, Contractor
Use Case: BBMP will assign pothole repair work to a Contractor. Further, the contractor will provide a cost estimate based on the size of the pothole. The software will automatically calculate the pothole size from photograph and cross check with the details submitted by Contractor. BBMP will approve the cost estimate and the contractor will perform the work assigned. Contracts will be assigned by area - constituency / wards / pin code. The contractor will be paid on a monthly basis.

2.4. UC04 - Work Progress and Tracking

Actors: Citizen, BBMP, Contractor
Use Case: Citizens/BBMP/Contractors can track their request with unique Request ID generated by the system. With navigation of potholes, their status will be provided. The

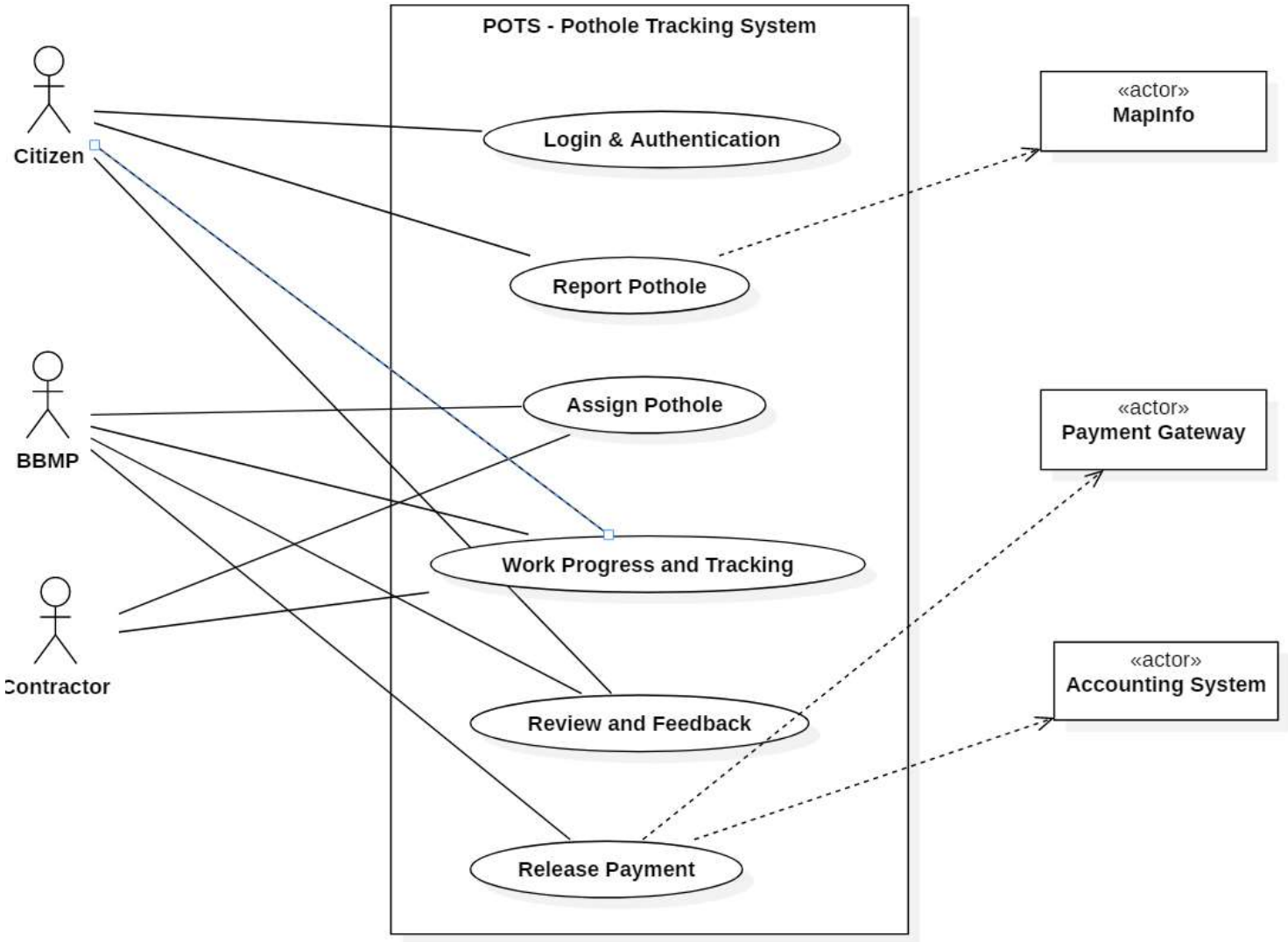
Mayor should be able to visualize the progress for overall city or by an area - constituency / pin code or by individual roads.

2.5. UC05 - Review and Feedback

Actors: Citizen, BBMP, Contractor
Use Case: BBMP can review work quality and provide feedback to the Contractor. Citizen who have reported pothole will be able to report their satisfaction on repair work, with review and feedback. BBMP can reassign work to the contractor (Or another Contractor).

2.6. UC06 - Release Payment

Actors: Contractor, BBMP
Use Case: BBMP pays the cumulative cost of all potholes fixed in a month to contractors that are marked as completed. Payment is initiated by the BBMP and are routed through a payment gateway. Upon successful confirmation from the payment gateway, the accounting systems of BBMP are updated for reporting expenses



Partial use-case context diagram

3. Fully Dressed Use Cases

3.1. UC1: Login and Authentication

Scope: Pothole Tracking and Repair System Login
Level: User goal
Primary Actors: Citizen, BBMP, Contractor

Stakeholder and Interests:
-Citizen, BBMP, and Contractor: wants fast and easy login/logout process. While capturing personal details minimum required fields should be mandatory. Easily able to tag the location of pot hole with longitude and latitude in the application.
-System: should provide easy process for reset and forget Password, access to camera and google database and user-friendly multi-lingual interface.
-BBMP: wants admin rights to modify any user request data or user data itself.
- Authentication: should be fast and always return correct result.
Preconditions: User profile must be created for successful login/logout
Success Guarantee: User is logged into the system and the based on the role of the user, services and features are made available to the user.

Main success scenario (MSS):

1. User accesses the URL/APP.
2. The system prompts the user for their login page/display.
3. The user enters his/her credentials - UserId and Password.
4. The system authenticates the user.
5. System display message: “Successfully logged-in”
6. The user gains access to the systems functionality.
7. Users can logout by using logout button.
8. System logged out and display popup message: “Successfully logged out”

Extensions (or Alternative Flow):
*a User profile creation.

1. For a new user, the system should provide an option for creating a new profile.
2. User name must be unique.
3. User shall input a user name of his/her choice.
4. User can choose any password of his/her choice.
5. User can verify the password by typing it again.
6. System checks, if the user name is not already in use.
7. System checks, if the two passwords are identical
8. System registers the new user with the credentials generated such as user name, password

6a. User name is already in use

1. User is requested to select another user name and password

7a. The two passwords are different

1. User is requested to retype (twice) his/her password

9. All personal data required to fill while creating new user profile. Name (First, Last & Middle), Mobile No, Full Address, Email-ID and DOB should be mandatory fields.
10. Location service if being used in app or web page should provide state, locality automatically during new profile creation.

Extensions

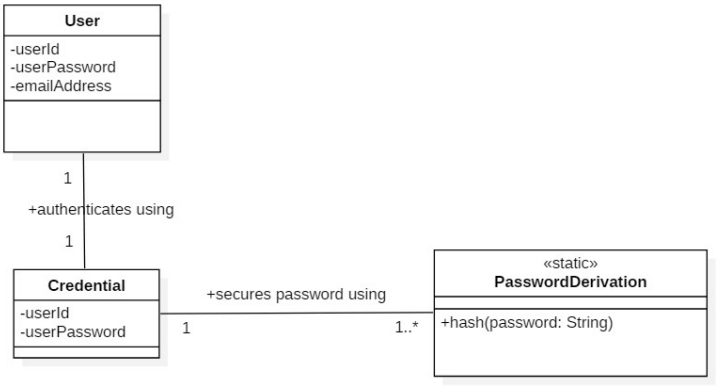
4b. Reset password.

1. System should provide reset password functionality is user forget password.
2. If reset password is requested, a level of authentication must be required, such as OTP on mobile or reset link in e-mail of user, as the case like.

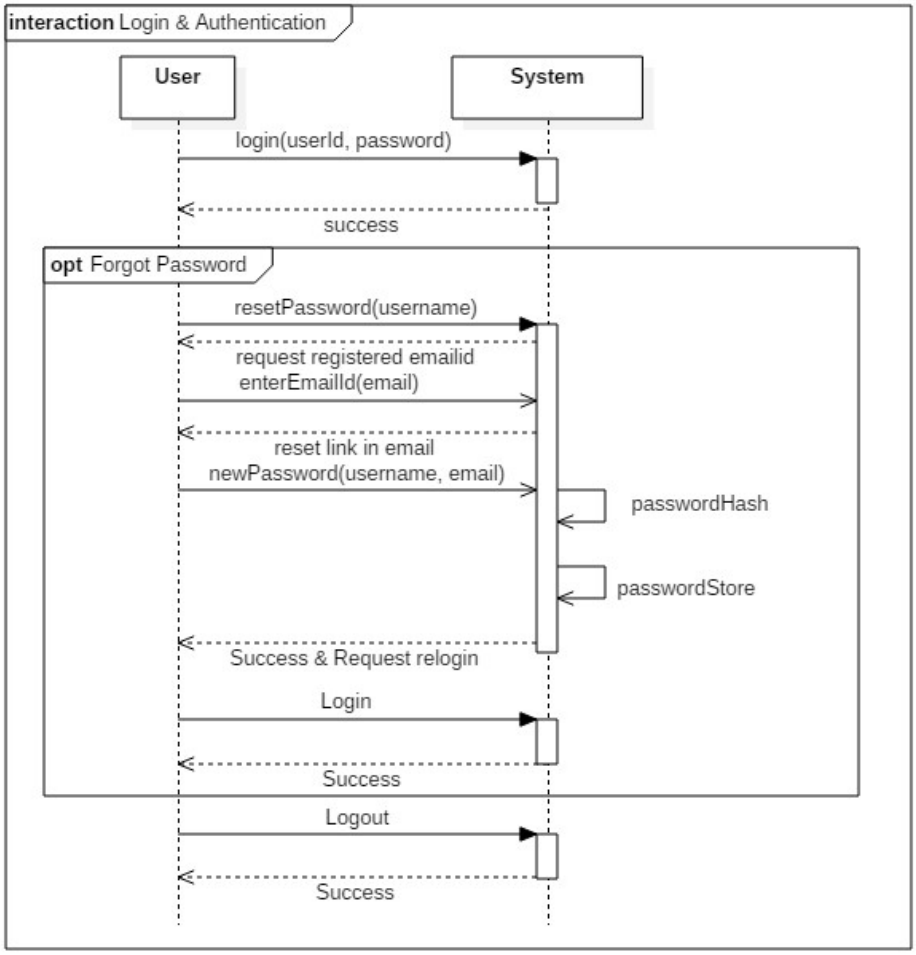
4c. BBMP admin rights

1. BBMP user can see the registered users, and data will be modifying using admin rights

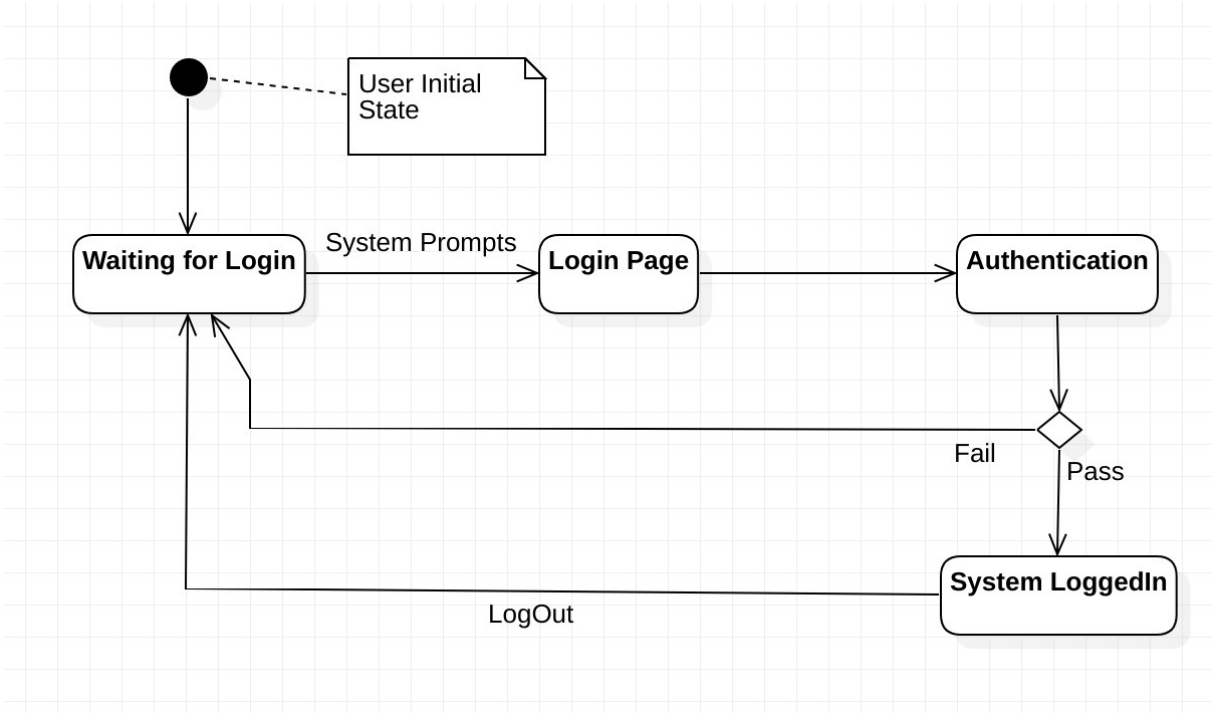
Class Diagram:



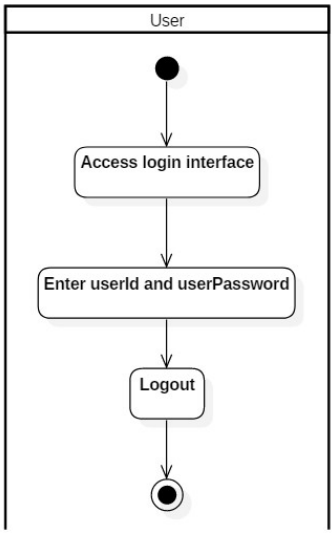
SSD:



State chart Diagram:



Activity Diagram:



1.1.UC2: Report Pothole

Scope: Pothole Report
Level: User goal
Primary Actors: Citizen
Stakeholder and Interests:

- Citizen: wants easy option to report pothole, map should be integrated while reporting. After report a unique ID should be generated for tracking
- System: should identify unique request and assigned a unique ID for identification of pothole.

Preconditions: Citizen must be logged in.

Success Guarantee: After successful creation of request a unique ID for the pothole, potholeId, should generated.

Main success scenario (MSS):

1. Citizen require adding pothole details like address, landmark, GPS location, photos, and pothole criticality.
2. The citizen needs to tag GPS location of pothole on Google Map.
3. The citizen needs to create request by clicking a submit button
4. System will generate unique ID.

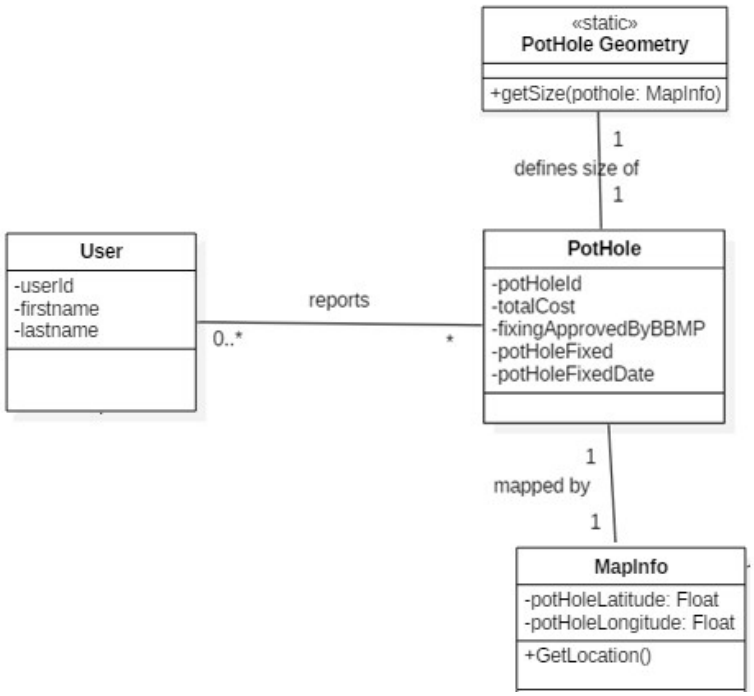
Extensions (or Alternative Flow):

*a. If system logged out user, System will system shall allow login again.

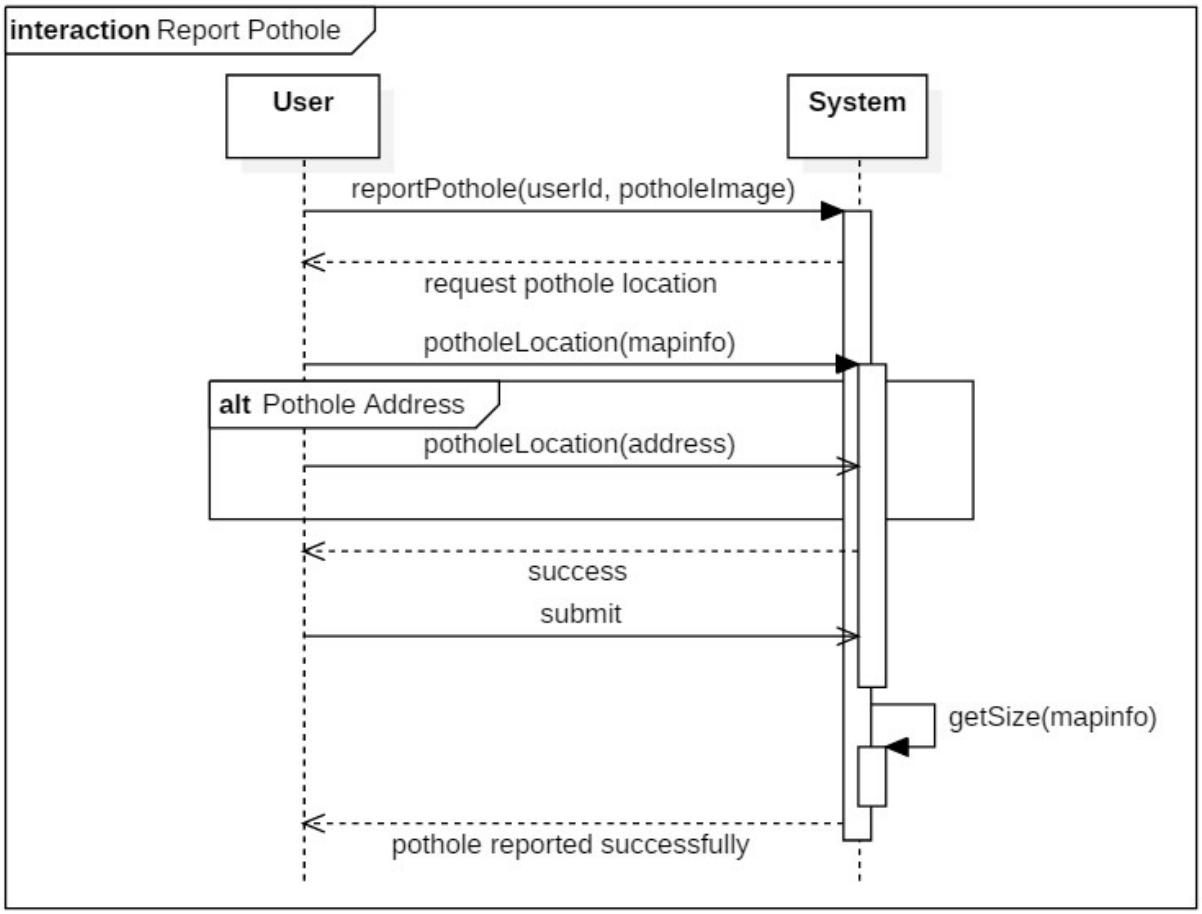
*1a2a. In case the citizen cannot locate the pothole on the Google maps service, he/she shall have the ability to enter the pothole address (including pincode).

3a. User can save its request and submit later. User can re-login and again open it older saved request and submit.

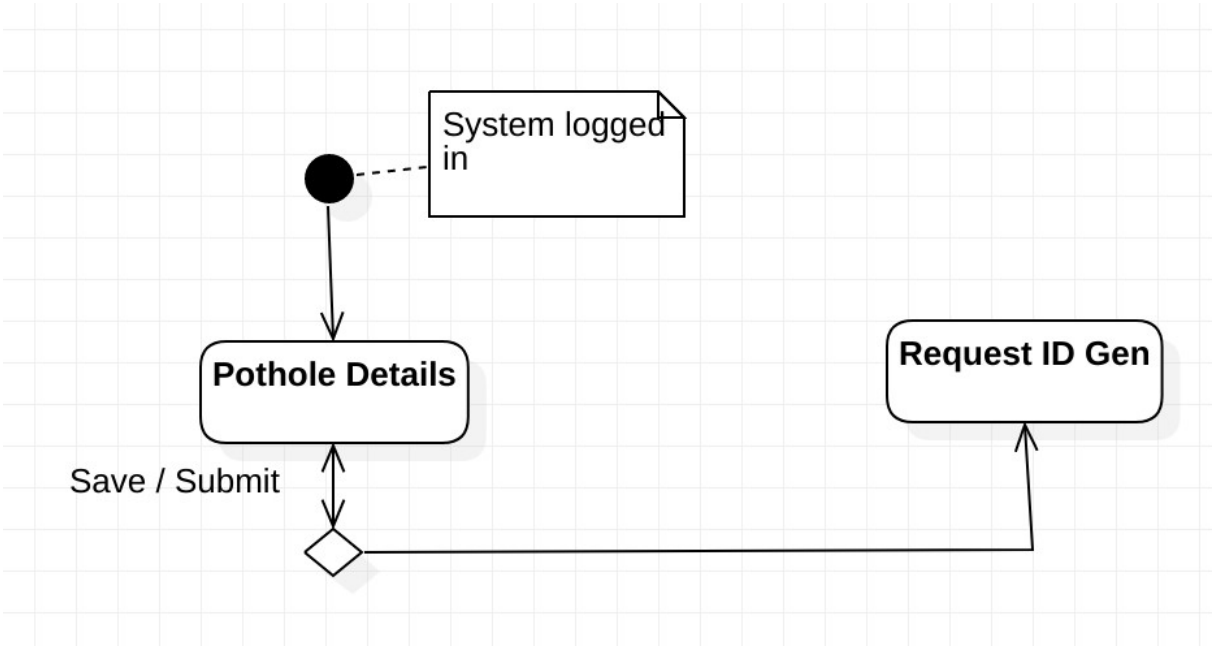
Class Diagram:



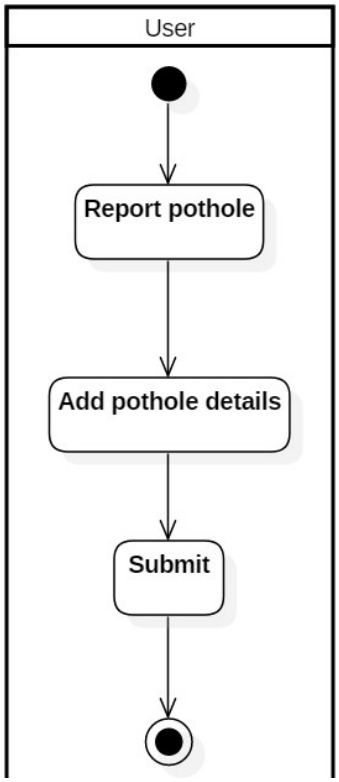
SSD:



State chart Diagram:



Activity Diagram:



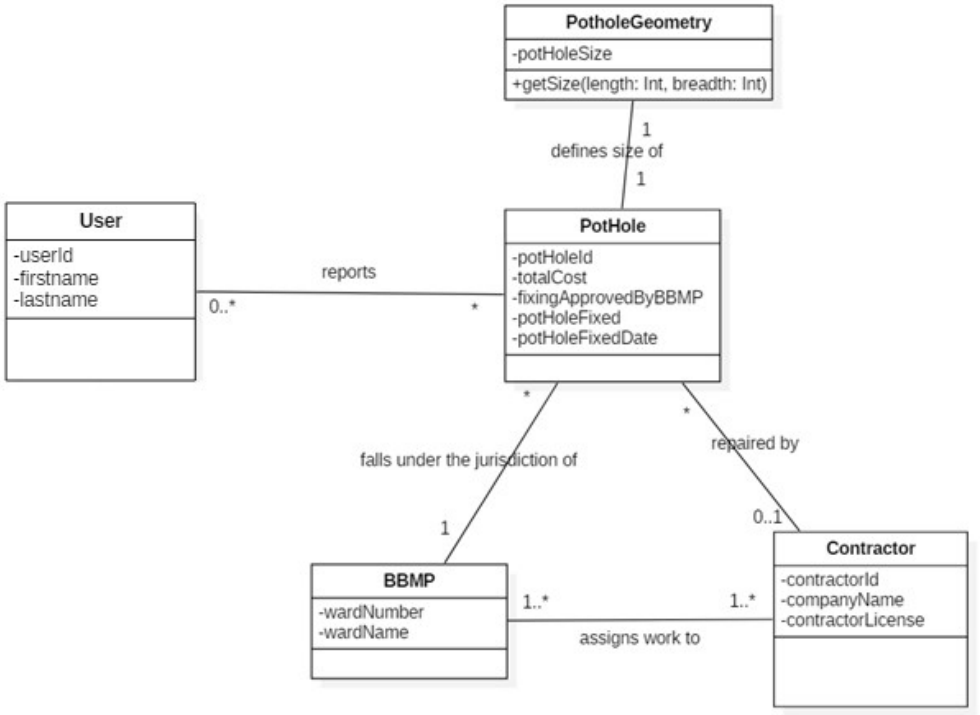
1.2. UC3: Assign Pothole

Scope: Assign Pothole
Level: User goal
Primary Actors: BBMP, Contractor
Stakeholder and Interests:
-BBMP: With unique ID BBMP can assign work to contractor.
-System: System shall able to calculate size of pothole from photos.
-Contractor: Shall able to provide cost estimation of pothole
Preconditions: Pothole request must be created, and unique ID must be generated. Valid contractor must be available.
Success Guarantee: Work will be assigned to contractor.
Main success scenario (MSS):

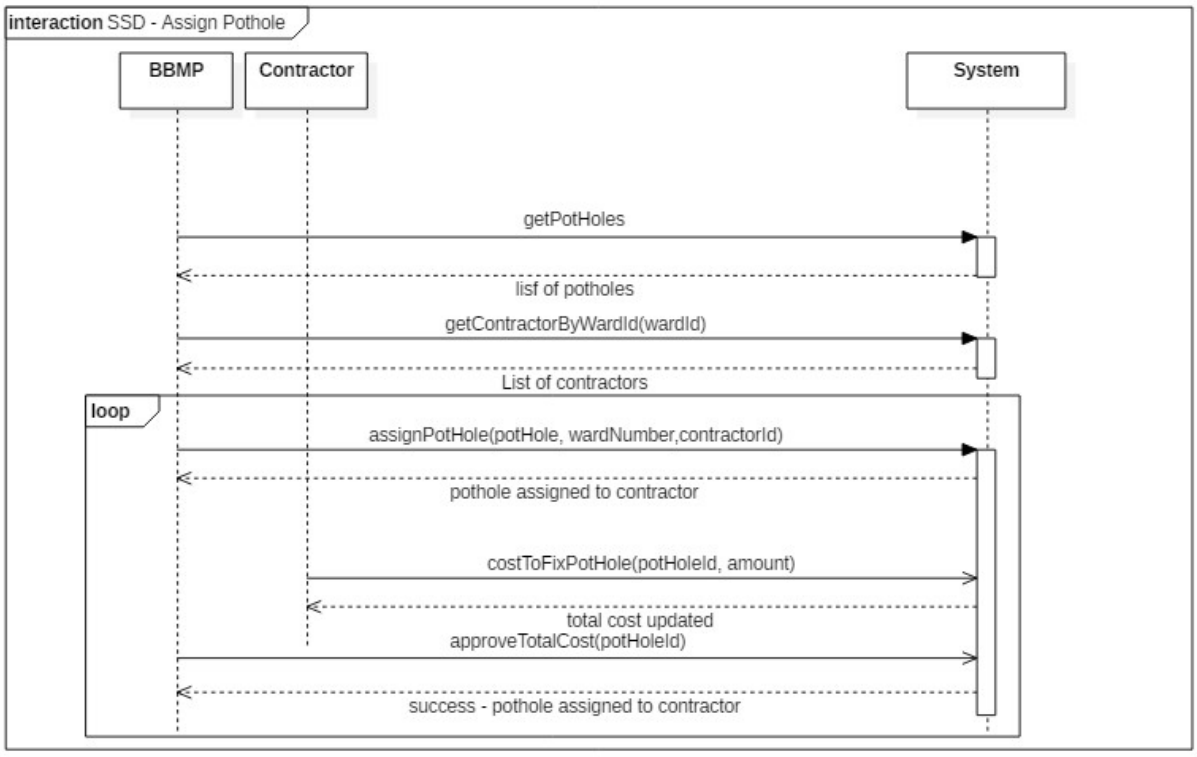
1. BBMP will login to system and check for new pothole request
2. New request will be assign to Contractor on basis of constituency / wards / pin code of request pothole.
3. Contractor will login to get new assignment.
4. Contractor will provide cost estimate based on the size of the pothole.
5. System will calculate cost on basis of size and photos.
6. BBMP will review both cost and approve work and cost to Contractor.

Extensions (or Alternative Flow):
*a. If system logged out user, System will allow again to login.
2a. In case of contractor not found in constituency / wards / pin code, BBMP will request to new contractor.
5a. In case of system not able to calculate cost of assignment, BBMP can fix cost for work assignment.
6a. In case cost provided by contractor is not approved by BBMP, BBMP can re-assign work to other contractor.

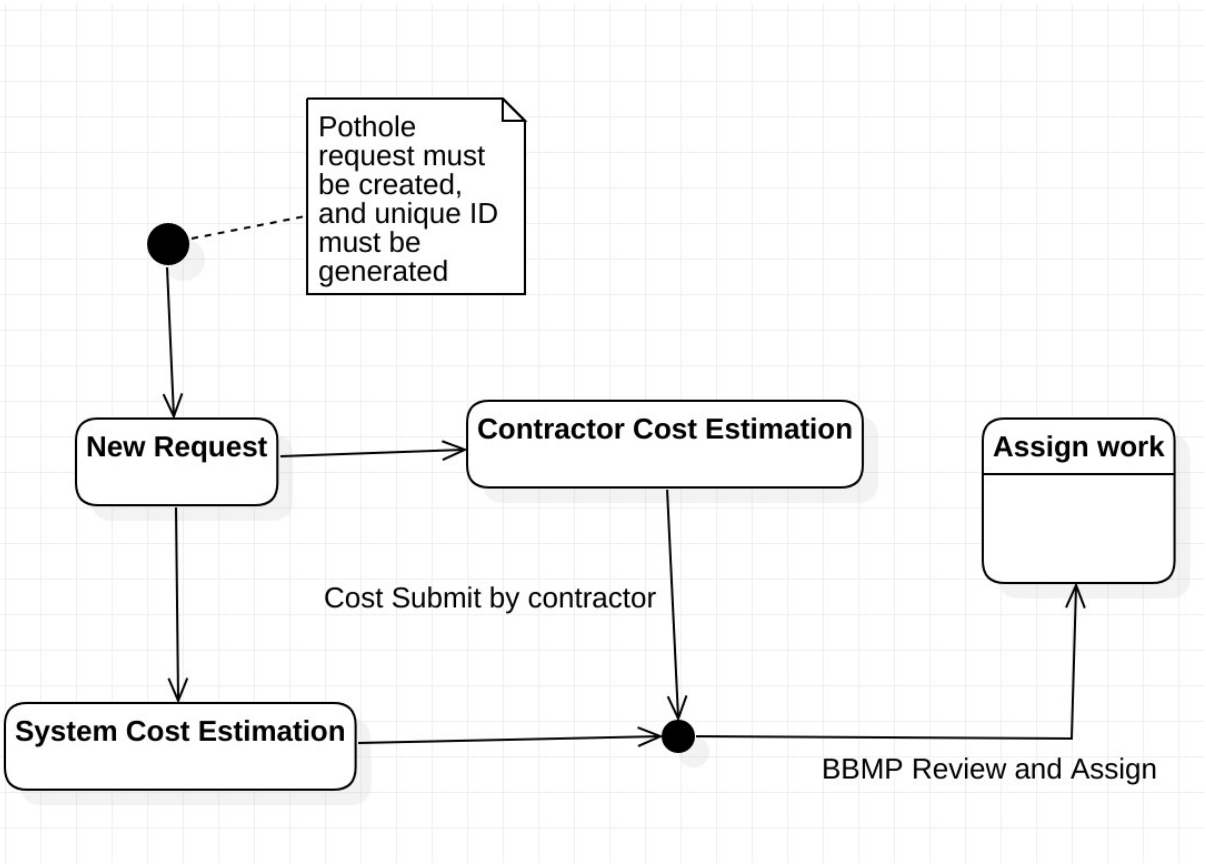
Class Diagram:



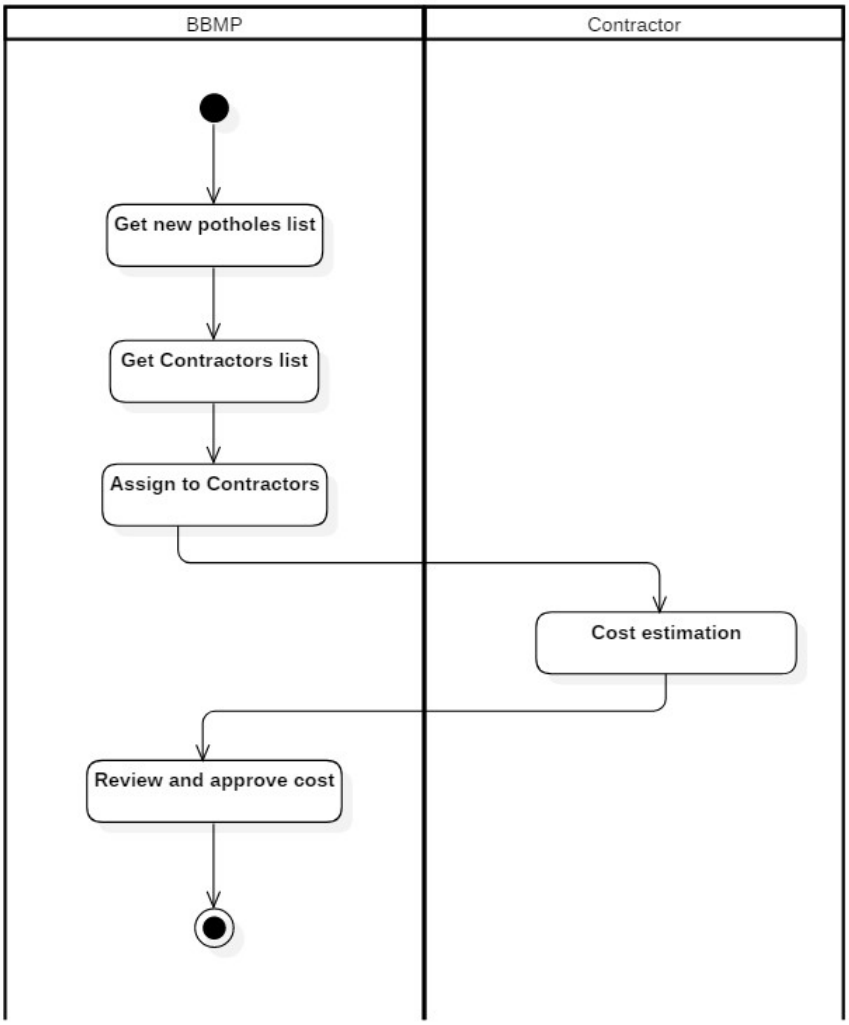
SSD:



State chart Diagram:



Activity Diagram:



1.3. UC4: Work Progress and Tracking

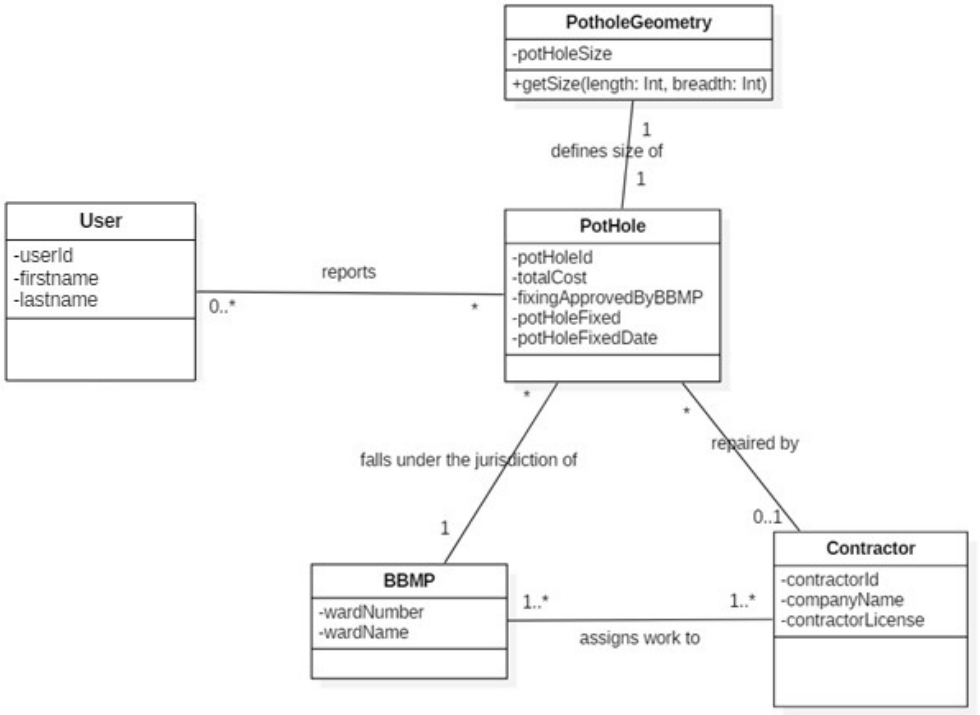
Scope: Work Progress and tracking
Level: User goal
Primary Actors: Citizen, BBMP, Contractor
Stakeholder and Interests:
-BBMP: With unique ID BBMP can track assigned work to contractor.
-System: System shall able to navigate pothole and its status
-Contractor: Shall able to get status of its work.
-Citizen: With unique ID Citizen can track work to progress.
Preconditions: Pothole request must be created, and unique ID must be generated. Work must be assigned to contractor.
Success Guarantee: Work progress can be tracked by User/BBMP/Contractor.
Main success scenario (MSS):

1. Citizens/BBMP/Contractors need to login to get tracking of work.
2. Citizens/BBMP/ Contractors need to enter unique request ID.
3. System will give information related to unique request ID
4. System will give navigation of potholes, their status.
5. System must allow to see all work assignment by an area - constituency / pin code or by individual roads.

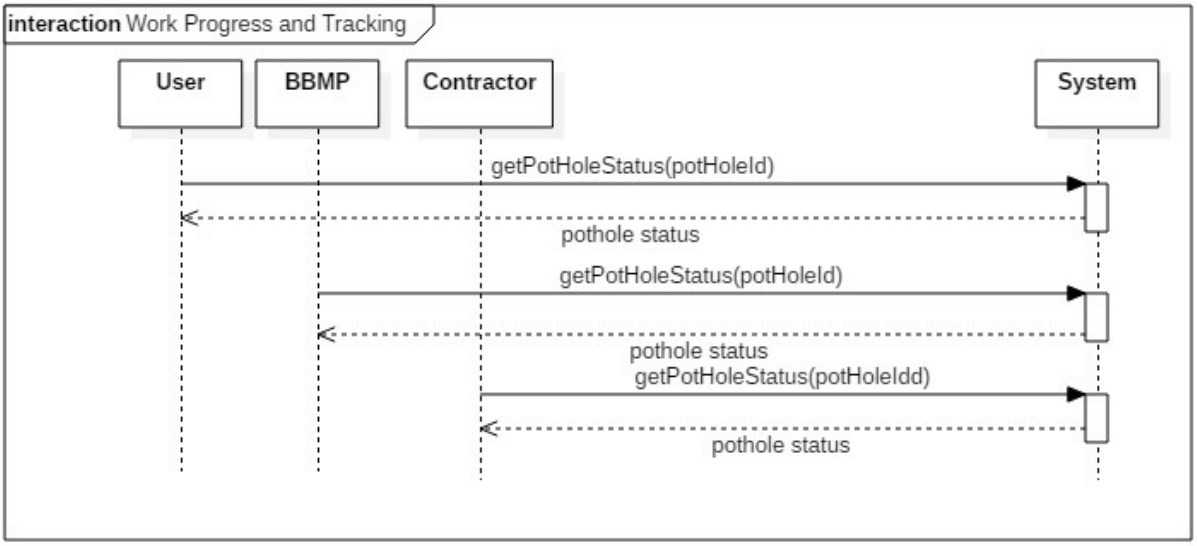
Extensions (or Alternative Flow):

*a. If system logged out user, System will allow again to login.
2a. If request ID is not valid system prompt error message. And re-allow to enter new request ID.

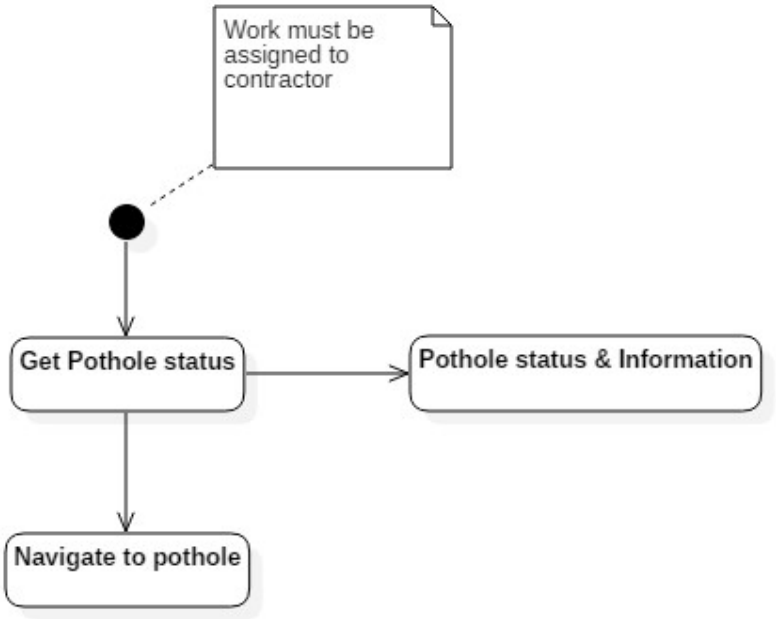
Class Diagram:



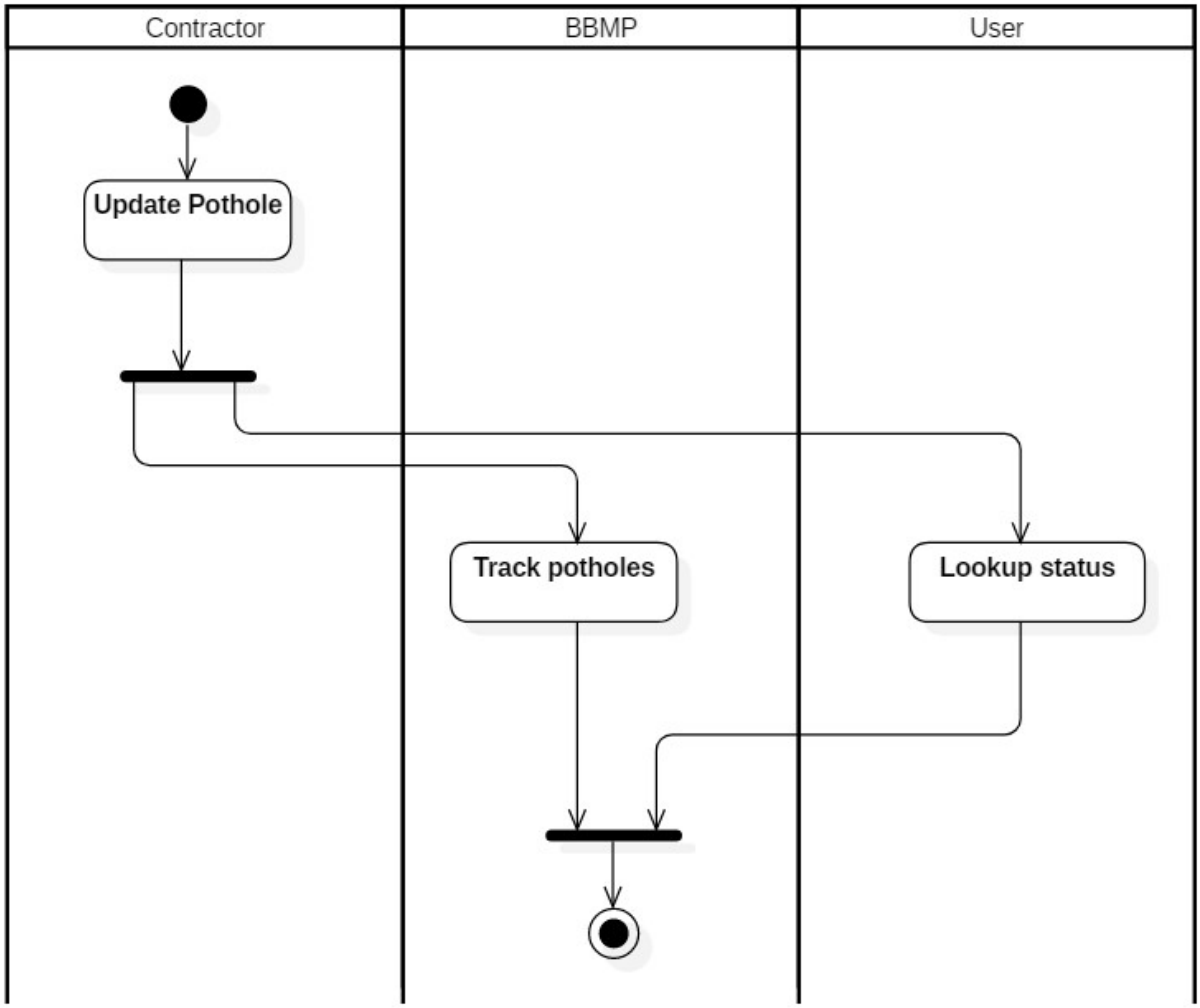
SSD:



State chart Diagram:



Activity Diagram:



1.4. UC5: Review and Feedback

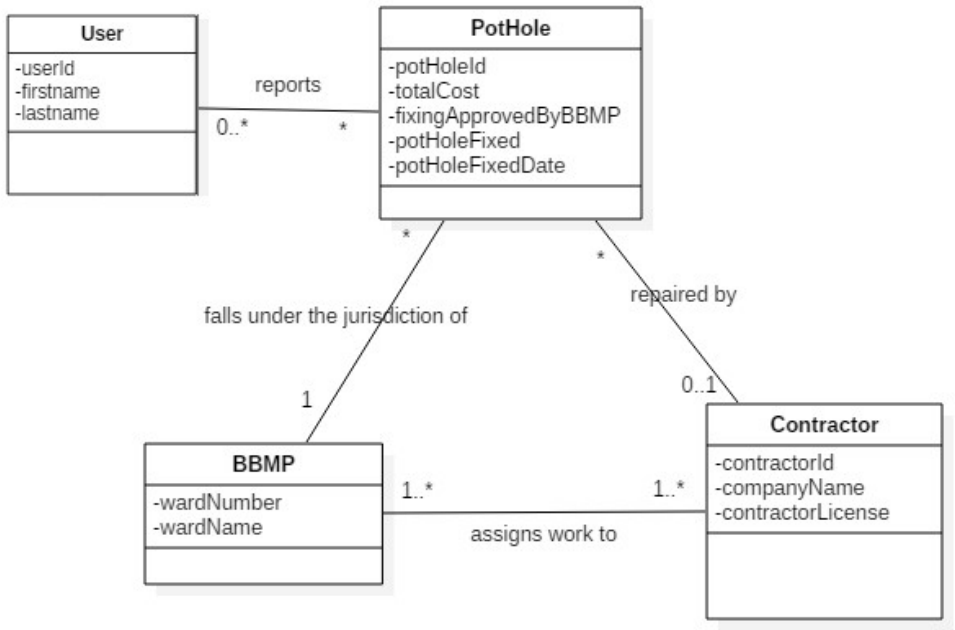
Scope: Review and Feedback

Level: User goal
Primary Actors: Citizen, BBMP, Contractor
Stakeholder and Interests:
-BBMP: With unique ID BBMP can check quality of work and feedback to Contractor.
-Citizen: With unique ID Citizen can review its request and feedback.
Preconditions: Pothole request must be created, and unique ID must be generated. Work must be assigned to contractor and finished.
Success Guarantee: Work review and feedback can be given by Citizen/BBMP.
Main success scenario (MSS):

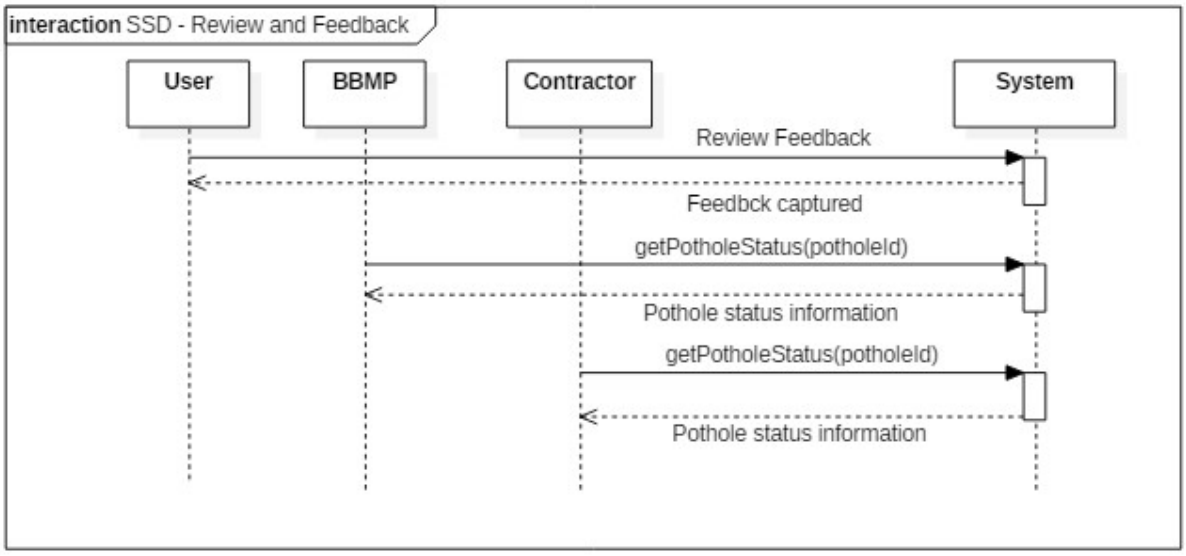
1. Citizens/BBMP/Contractors need to login to get work review status and feedback.
2. Citizens/BBMP/ Contractors need to enter unique request ID.
3. System will provide feedback and review window to enter comments and rating.
4. System will provide feedback and review window to contractor to see its work status.
5. System must allow re-assigning work to other Contractor.

Extensions (or Alternative Flow):
*a. If system logged out user, System will allow again to login.
2a. If request ID is not valid system prompt error message and re-allow entering new request ID.

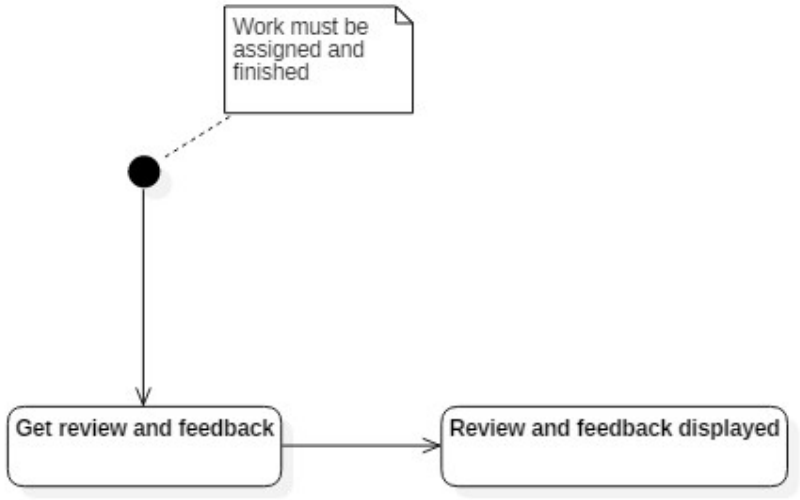
Class Diagram:



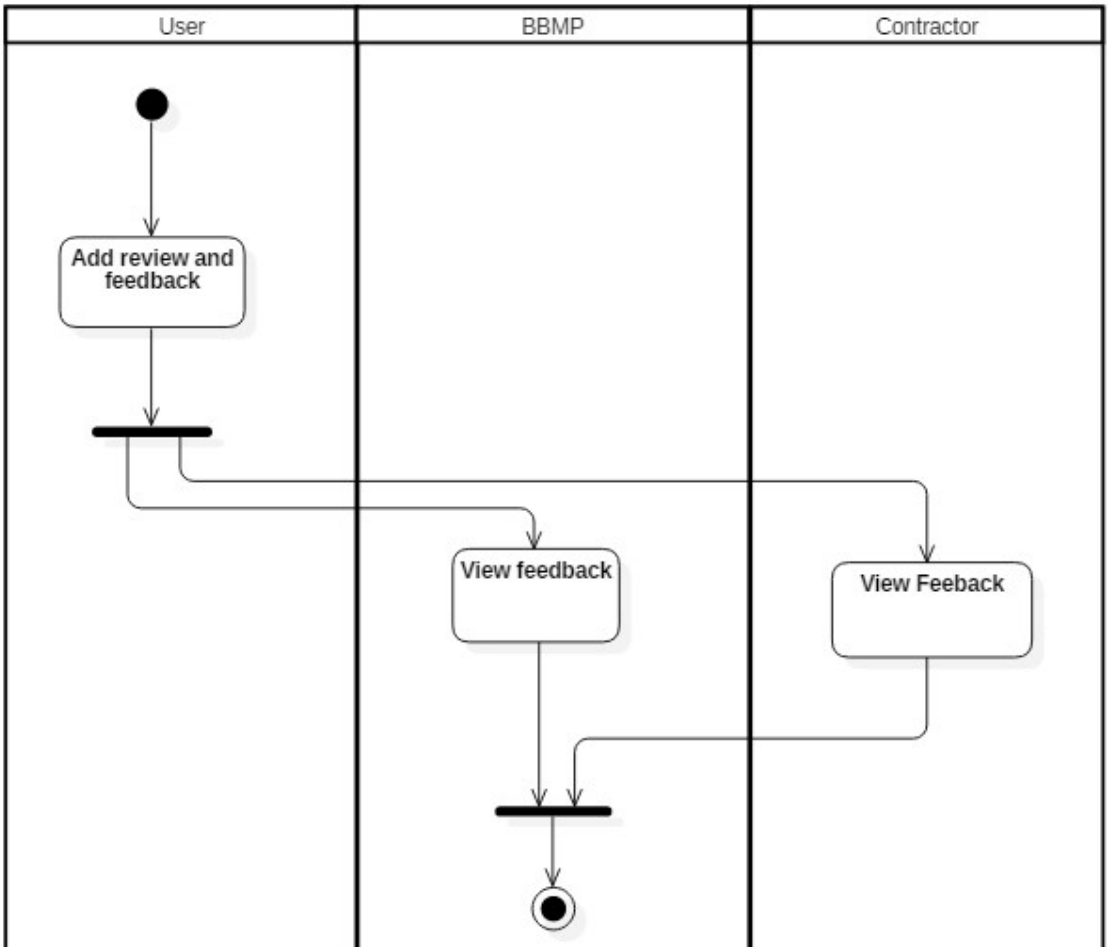
SSD:



State chart Diagram:



Activity Diagram:



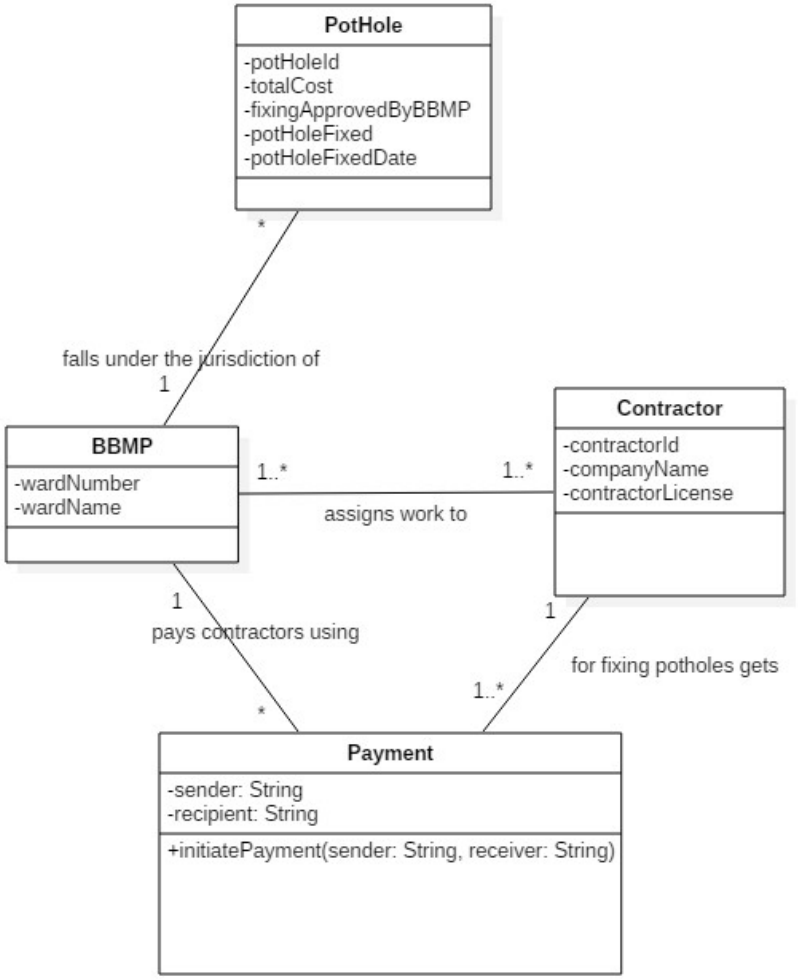
1.5. UC6: Release Payment

Scope: Release Payment
Level: System goal
Primary Actors: BBMP
Stakeholder and Interests:
-BBMP: Payment release for all assigned work packages to Contractors on monthly basis.
Preconditions: Assigned work to Contractor must be completed and reviewed by BBMP.
Success Guarantee: Payment will be release to Contractor.
Main success scenario (MSS):

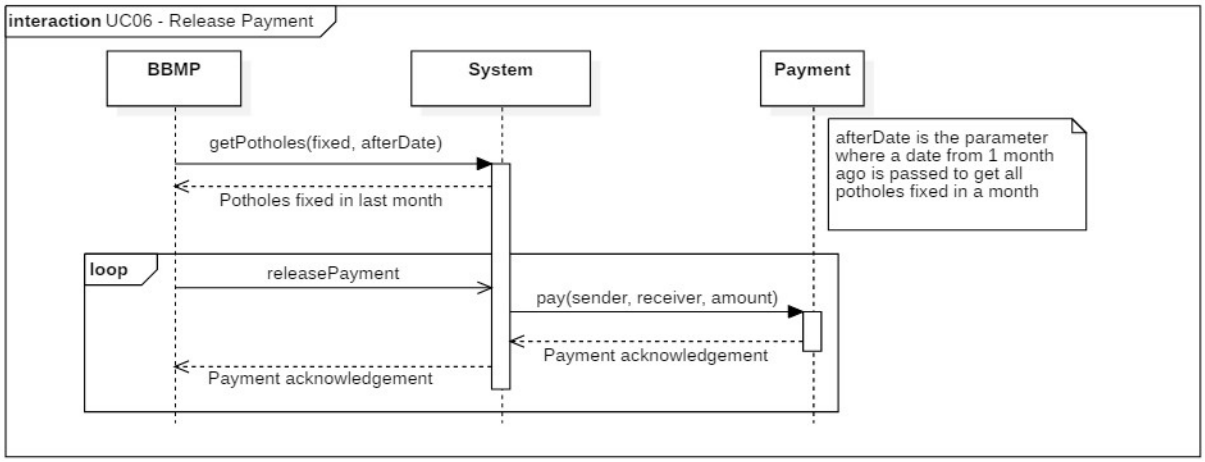
1. BBMP will identify all work packages completed in current month.
2. Estimate overall cost of all completed work packages.
3. Payment will be initiated to respective Contractor using payment gateway.
4. On acknowledgment of payment work will be mark as closed
5. System will generate monthly expanse report.

Extensions (or Alternative Flow):
2a. In case of payment failure, system will prompt User to make payment again.
4a. If no acknowledgment received for payment, work package status will mark as pending for payment.

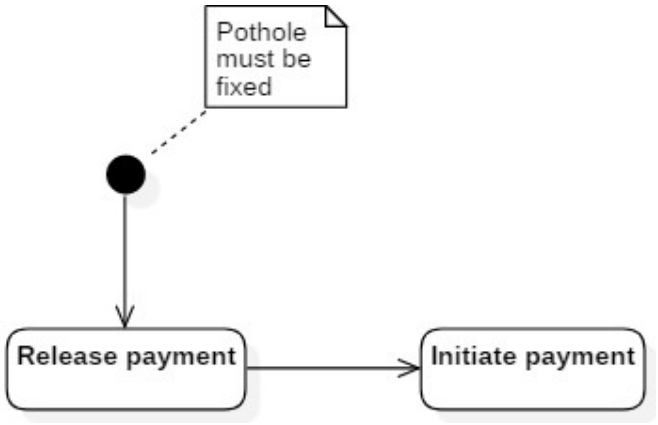
Class Diagram:



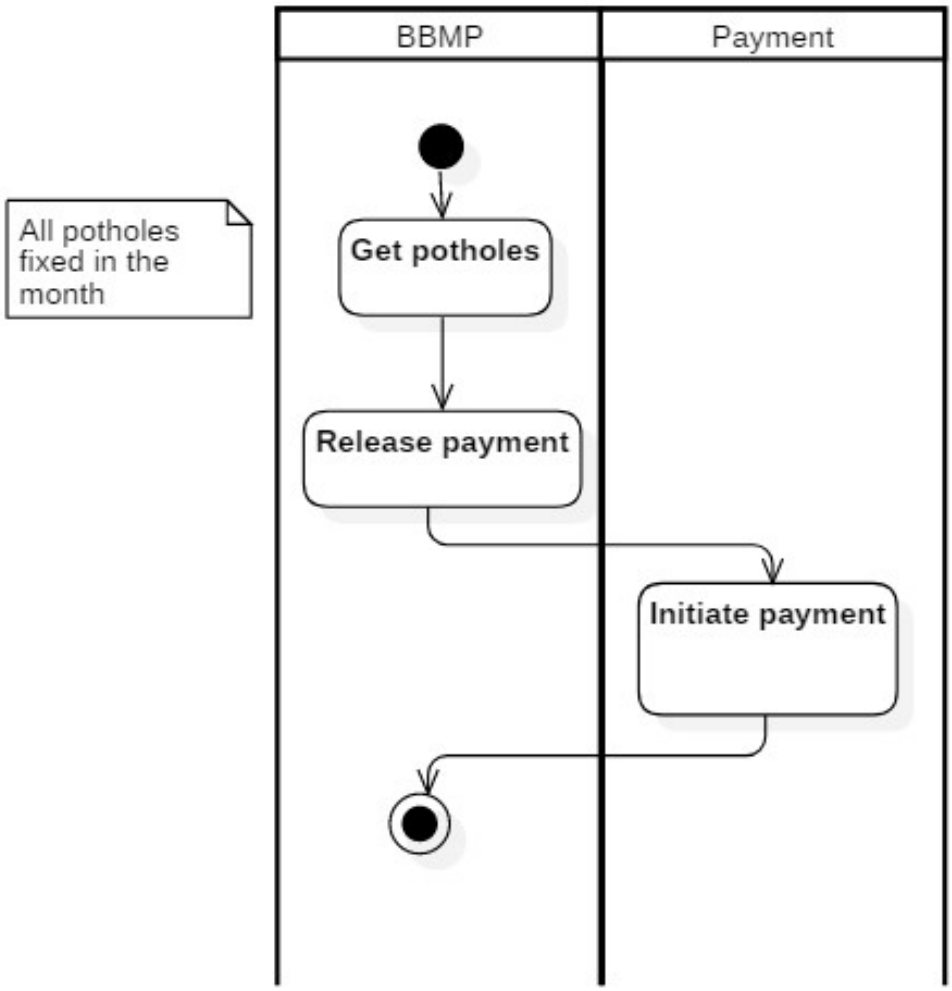
SSD:



State chart Diagram:



Activity Diagram:



2. System Contracts

2.1. Contract C01: Login

Operation: login (username:String, password:String)
Cross References: Use case: UC01: Login and Authentication
Preconditions: A user is registered
Postconditions:

- User credential object created.
- Password is hashed using the systems' hashing function
- User provided password hash is compared to the one stored in the system
- User logged on if passwords match
- Previous reported pothole information is loaded if platform is mobile

2.2. Contract C02: Report Pothole

Operation: reportPothole(userId: Int, address:String, potholeImage:Binary)
Cross References: Use case: UC02: Report Pothole
Preconditions: A user is registered and logged in
Postconditions:

- A new pothole object was created
- Pothole status set to ***Reported***.
- Pothole GPS coordinates was captured from Map metadata
- Pothole associated with a ward where the work must be done
- Pothole size was calculated and added to the pothole object
- Pothole associated with the user

2.3. Contract C03: Assign Pothole

Operation: assignPotHole(Pothole: pothole, wardnumber:String, contractorID:Int)
Cross References: Use case: UC03: Assign Pothole
Preconditions: Pothole request must be created, and unique ID must be generated. Valid contractor must be available.
Postconditions:

- A new contractor object was created
- Pothole status set to ***Assigned***.
- Ward Number update in contractor

2.4. Contract C04: Work Progress and Tracking

Operation: getPotHoleStatus(Pothole: pothole)
Cross References: Use case: UC04: Work Progress and Tracking
Preconditions: Pothole request must be created, and unique ID must be generated. Work must be assigned to contractor.
Postconditions:

- Pothole status set to ***Work in Progress***.
- Status of work, details of pothole will be return to user.

2.5. Contract C05: Review and Feedback

Operation: setReview(Pothole: pothole)
Cross References: Use case: UC05: Review and Feedback
Preconditions: Pothole request must be created, and unique ID must be generated. Work must be assigned to contractor and finished.

Postconditions:

- Pothole status set to ***Work finished***.

- Status of work, review and feedback window return to user.
- Feedback provided to contractor.

2.6. Contract C06: Release Payment

Operation: releasePayment(Pothole: pothole,contractorID:int)

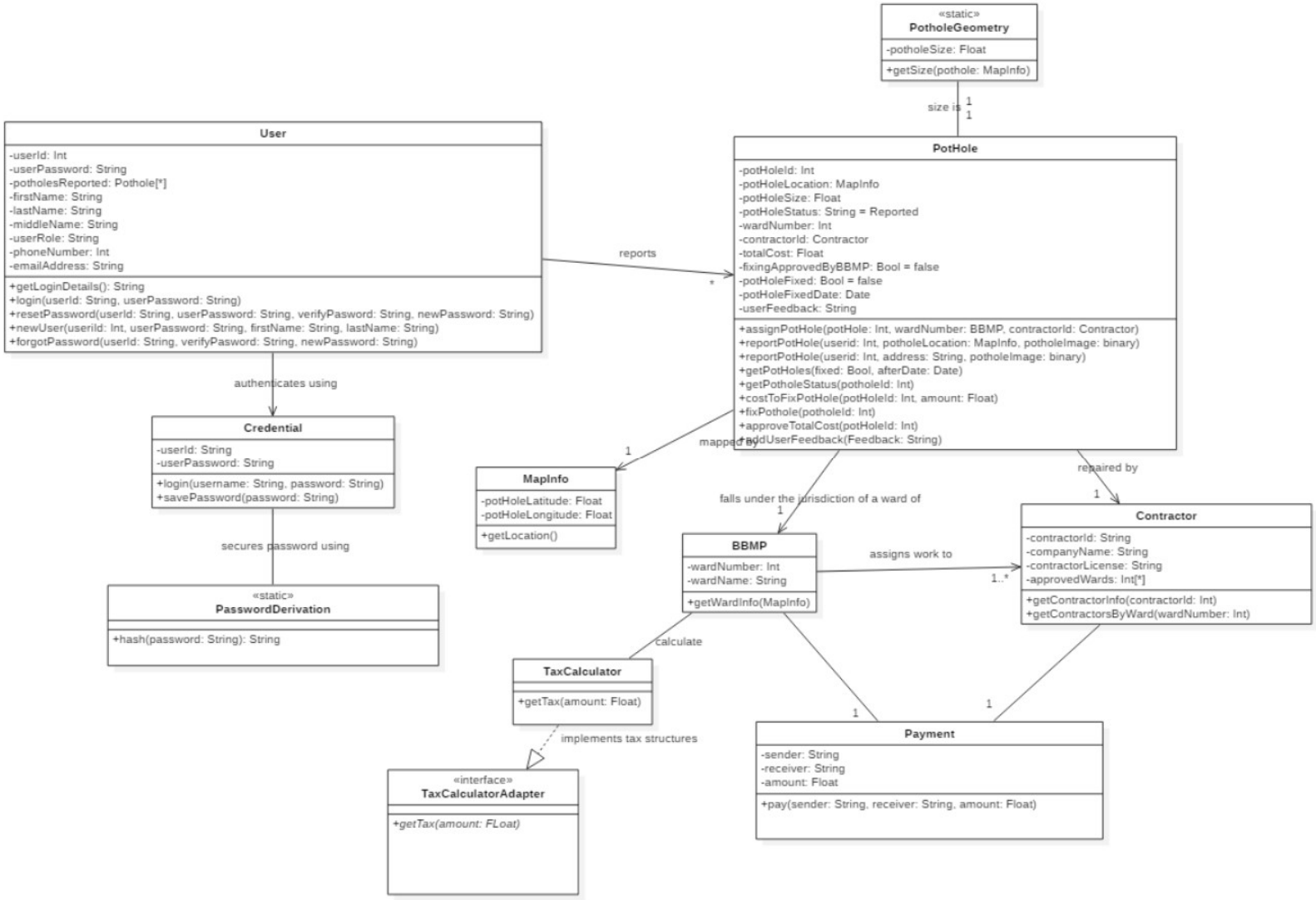
Cross References: Use case: UC06: Release Payment.

Preconditions: Assigned work to Contractor must be completed and reviewed by BBMP.

Postconditions:

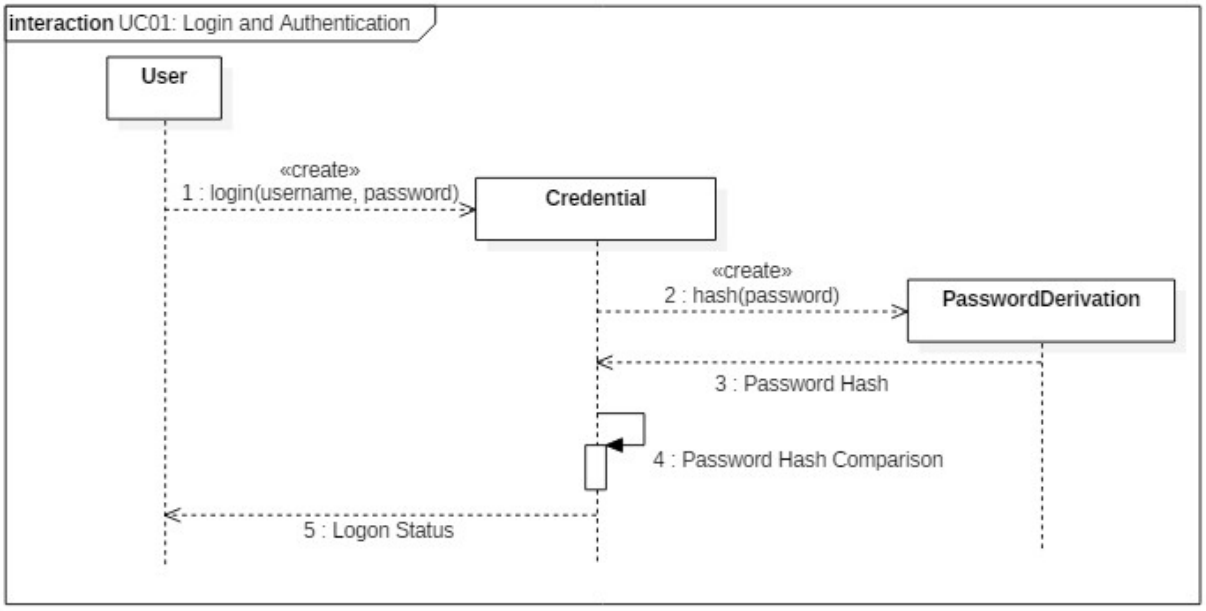
- Pothole status set to ***Work finished***.
- Payment will be released to contractor using payment gateway
- Assigned work mark as close

3. Design Class Model

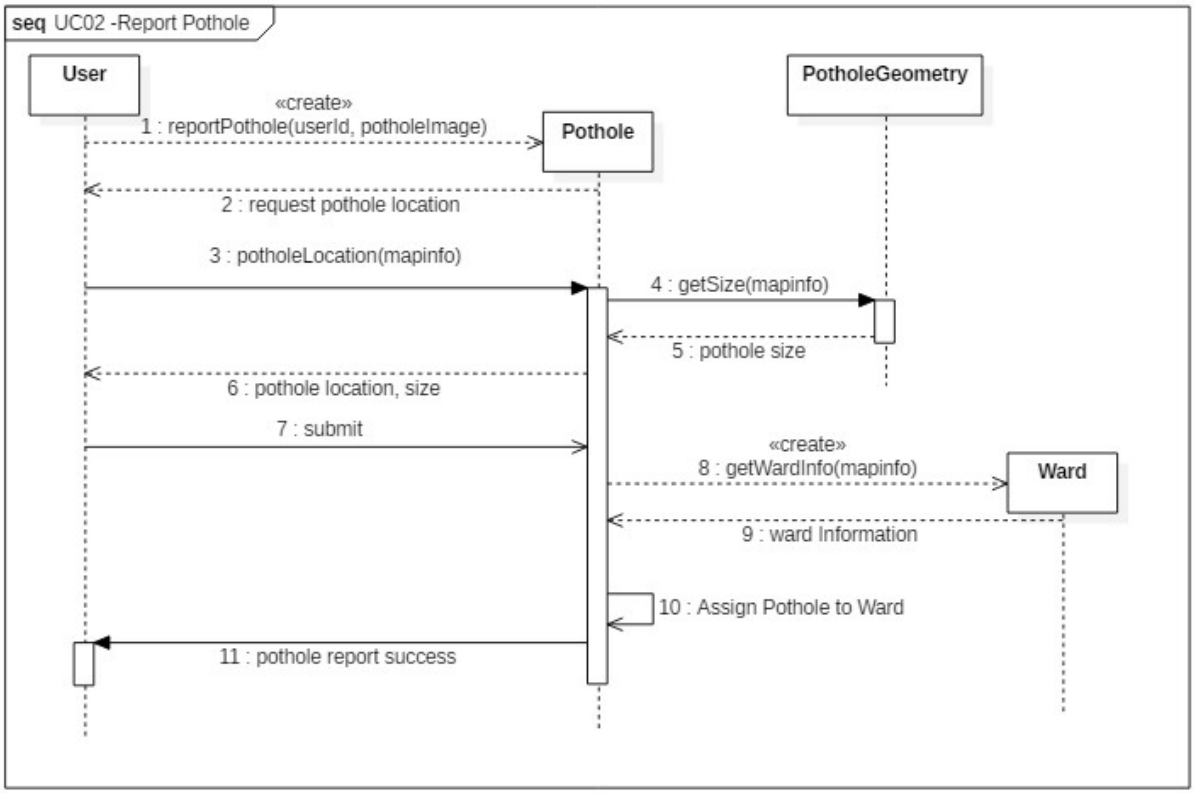


4. Interaction Diagram

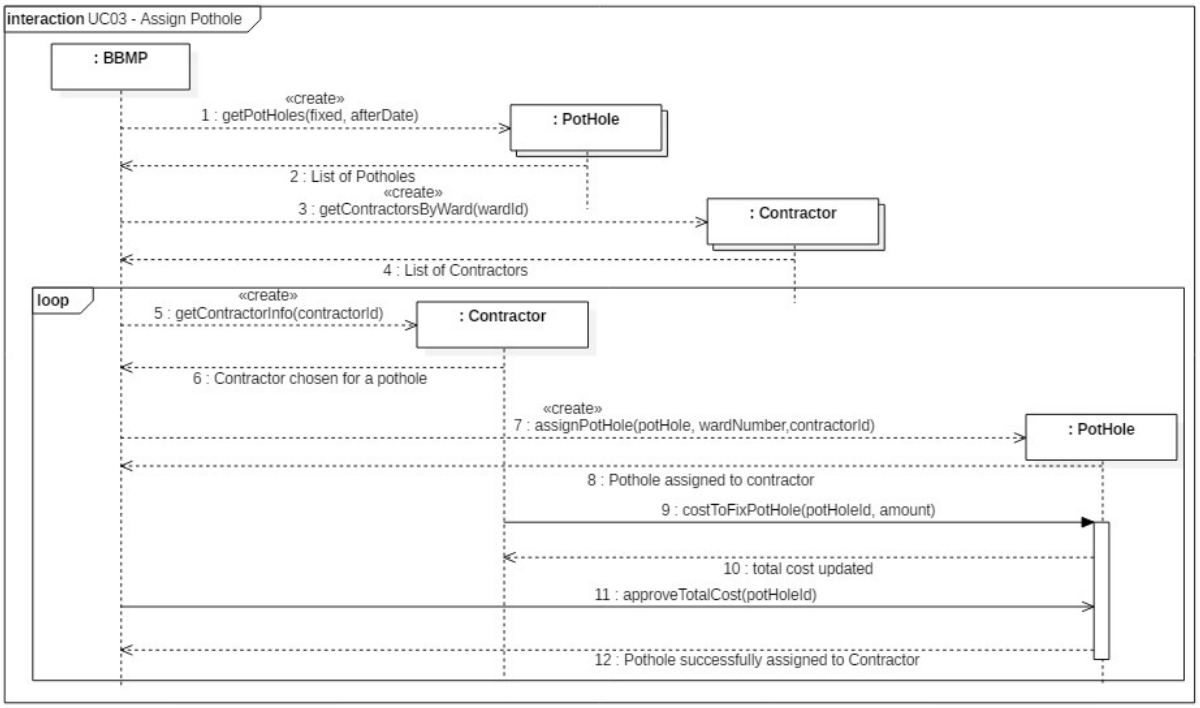
4.1. UC01-Login Interaction



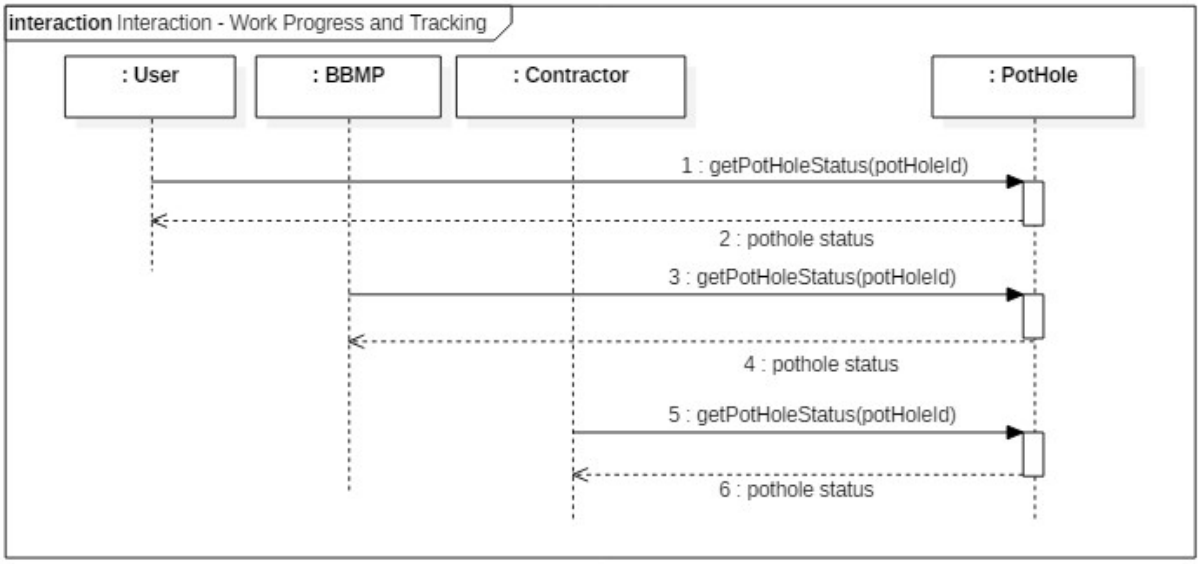
4.2. UC02-Report Pothole Interaction



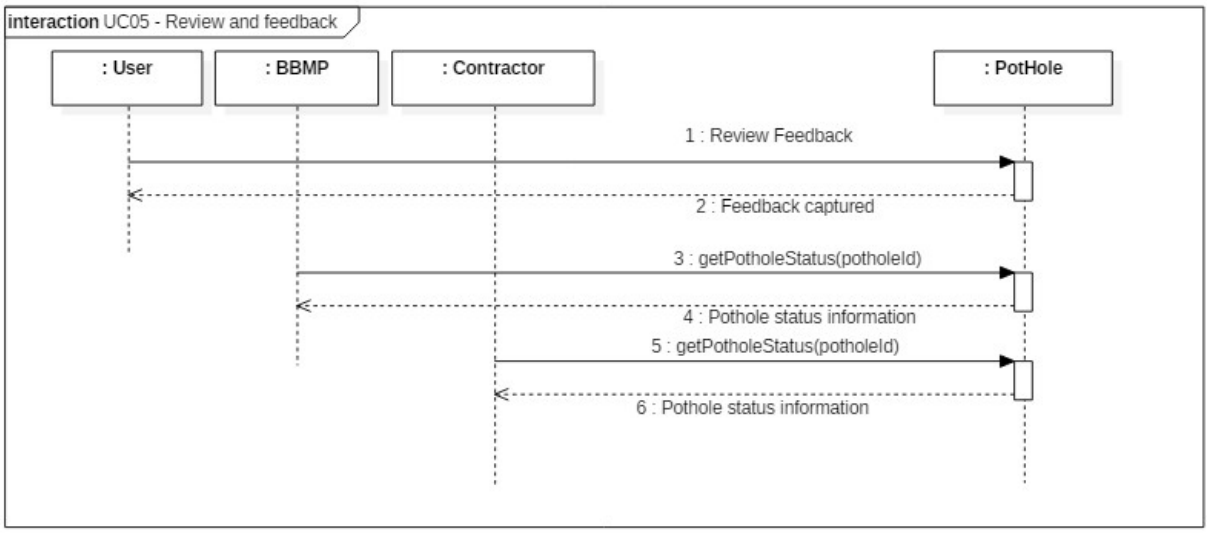
4.3. UC03-Assign Pothole Interaction



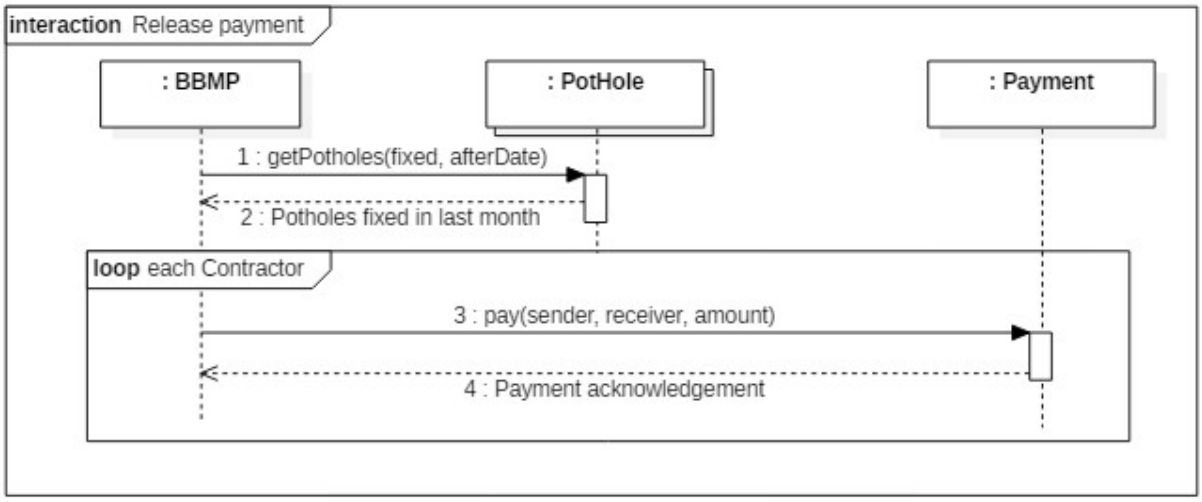
4.4. UC04-Work Progress and Tracking Interaction



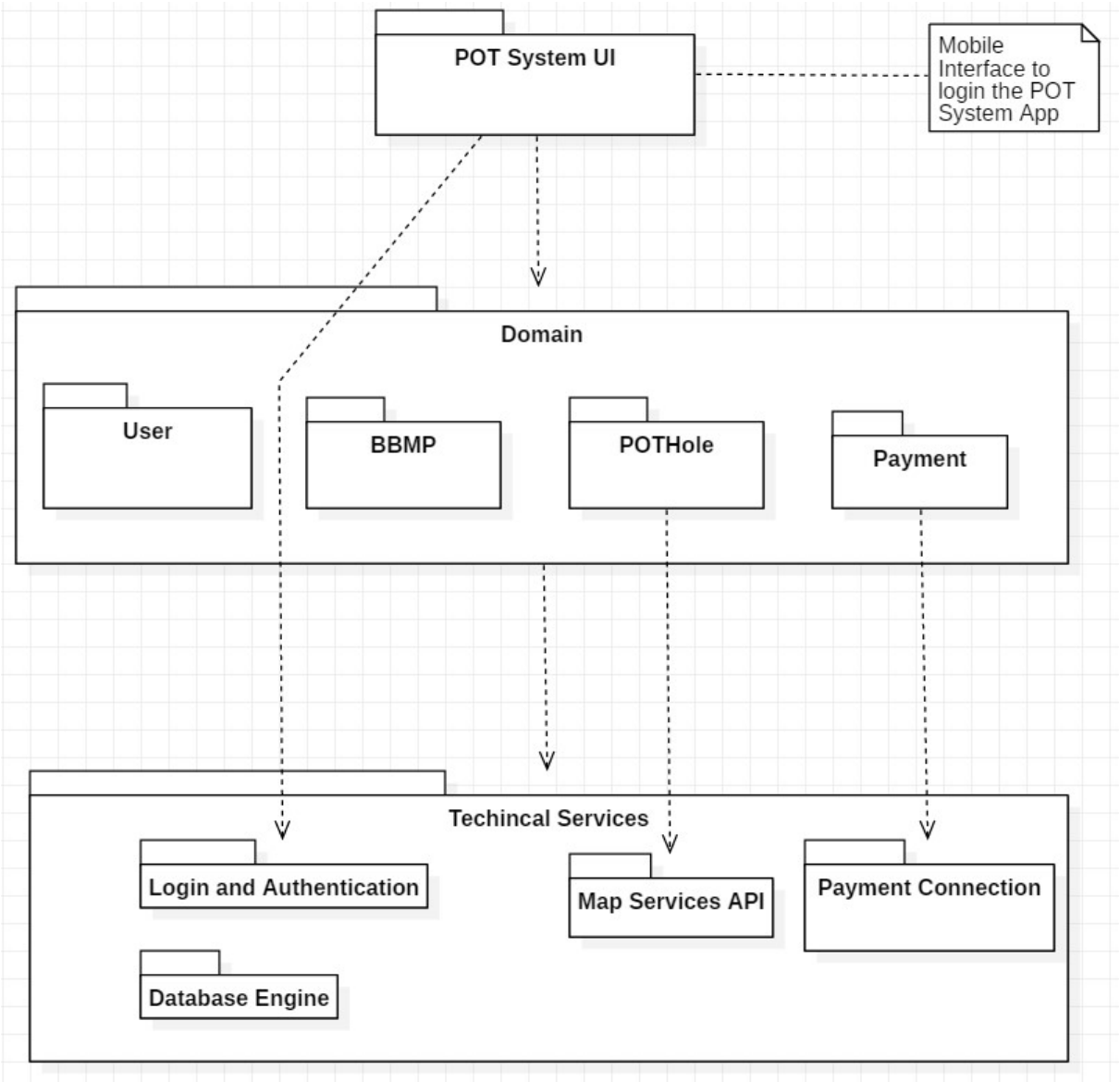
4.5. UC05-Review Work and Feedback



4.6. UC06-Release Payment

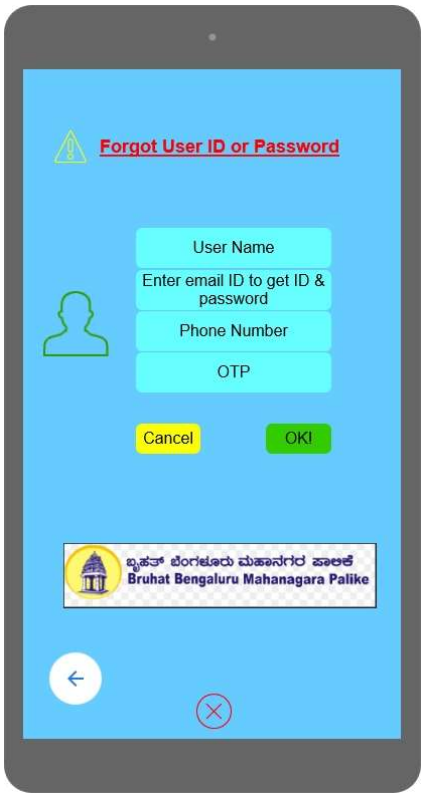


5. Package Diagram:



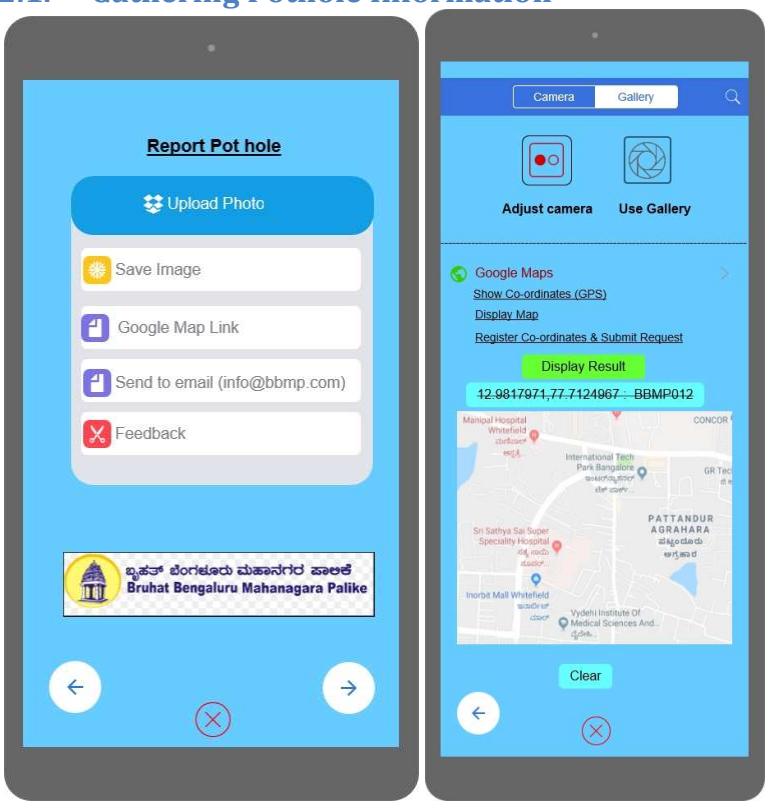
6. User Interface Mock Ups

6.1. Login screen

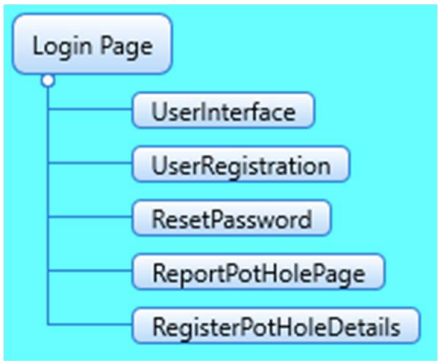


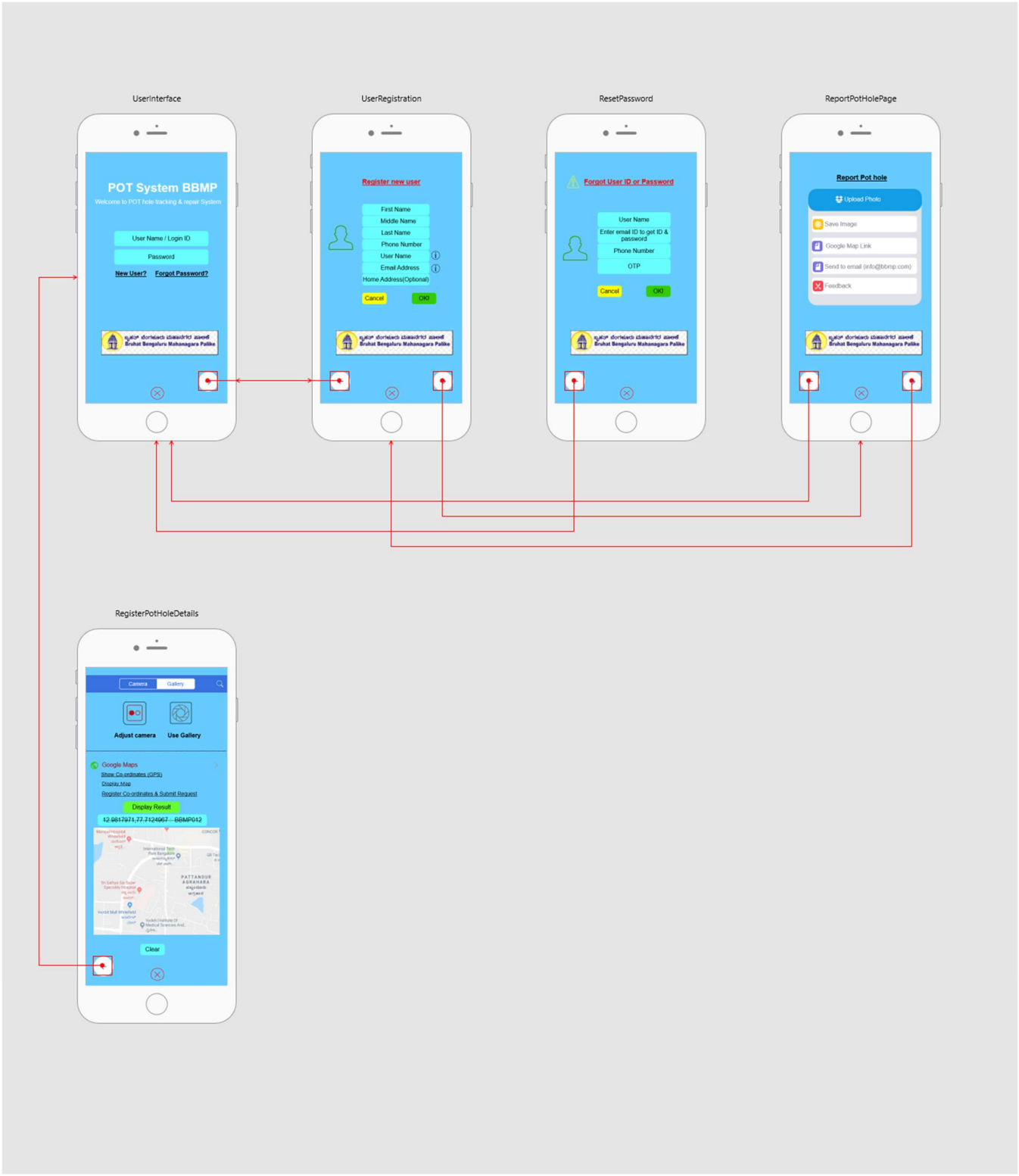
6.2. Report Pothole

6.2.1. Gathering Pothole Information



6.2.2. Flow Diagram of user interface





7. GRASP principle application

1) Creator:

-Assign class B the responsibility to create an instance of class A.
-In UC02 we have applied this principle where class User Assign responsibility to create object of pothole and class Credential.
Methods are inherited, So user can requests like getStatus, getTracking, review, login etc.

-Similarly class payment assign responsibility to create object of payment Gateway or Tax Calculator Adapter class.
-When pothole object has created then pothole class responsible to create PotholeGeometry class, but methods can be access from user class.

2) Controller:

-Assign the responsibility for receiving or handling a system event message.
-An input system event is an event generated by an external actor. They are associated with system operations. Operations of the system in response to system events, just as messages and methods are related.
- User and BBMP class are controller class.
-User class handling all external event like, user/password creation, login, report pothole etc.
-BBMS is controlling cost estimation, identify contractor on basis of ward number, assignee work packages.
-And releasing payment to contractor on completion of work.

3) Polymorphism:

-When related alternatives or behaviors vary by type (class), assign responsibility for the behavior—using polymorphic operations—to the types for which the behavior varies.
-In UC06 payment class use external payment Gateway class. When BBMP issue makePayment() event to payment class, payment class use Polymorphism to call all different type of payment gateways methods.
- getWorkStatus(PotHoleID) from BBMP, User, Contractor using Polymorphism, because each class expect different status. Like User class, want work progress, Contractor class want pending payment info etc.

4) Protected Variations:

-By creating <<interface>> class we can hide complex details of class.
-In our design payment <<interface>> class is used to hide complex and different type of payment gateway.
-Later if new payment way added, we can easily add into our design like BHIM, PhonePay etc.

5) Low Coupling:

Mobile Application POT system Interface class (System in interaction diagram of UC01) -> User Class ->Credential class -> Password Derivation Class.
(The mobile interface class create user object then the user object creates credential class object for password generation and uses password derivation object to generate password hash. The POT system interface does not know about password derivation class methods and objects.)

User Class ->POTHole Class -> PotHoleGeometry Class
User class object creates pothole class object and calls method to get the geometry of the pothole. The user class does not know about geometry class. Hence supports low coupling.

6) High Cohesion:

PotHole class is in high cohesion with geometry class, as they know their clear responsibility. Report pothole (System in interaction diagram of UC02) has clear responsibility of the work between user and system.

7) Information Expert:

Mobile Application POT (System in interaction diagram of UC01) Interface class aggregates all the information regarding user class and assigns task for user object creation.

8) Controller:

-Assign the responsibility for receiving or handling a system event message.
-An input system event is an event generated by an external actor. They are associated with system operations—operations of the system in response to
System events, just as messages and methods are related.
-In our design majorly User and BBMP class are controller class.
-User class handling all external event like, user/password creation, login, report pothole etc.
-BBMS is controlling cost estimation, identify contractor on basis of ward number, assign work packages.
-And also releasing payment to contractor on completion of work.

9) Indirection:

Payment gateway interface which extends a 3rd party in UC06 uses TaxCalculator Adapter (indirection) to solve the tax computation for payments.

10) Pure Fabrication:

Payment Gateway, the domain classes need to store information about the customers. In order to do that one option is to delegate data storage responsibility to domain classes.

8. Summary Document

8.1. Iteration 1, review comments and changelog

1. Functional requirements of Pothole Repair and Tracking describe the functionality of the system. They describe exactly what tasks the software must perform. Functional requirements(Use-Case) define the scope of the system,
2. The product boundaries, and its connections to adjacent systems or class. Use-Case also define the business rules like payment, work assigmant,cost estimation etc.
3. Business rules are the rules that the system must conform to, based on the individual task or work assignment. This includes defining the data or task that must be tracked.
4. The business rules are the most important type of functional requirements and most of your requirements will be of this type.
5. Use-Cases
   UC01 - Login and Authentication:
   UC02 - Report Pothole
   UC03 – Assign Pothole
   UC04 - Work Progress and Tracking
   UC05 - Review and Feedback
   UC06 - Release Payment
6. Convert UC01 and UC02 in fully dresses.
7. All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees, MSS, Extensions (or Alternative Flow).
8. Drive Domain Class Diagram/Conceptual Class of UC01 and UC02.
9. Learning
   1. Domain objects or conceptual classes
   2. associations between conceptual classes
   3. attributes of conceptual classes

8.2. Iteration 2, review comments and changelog

10. Added system contracts for methods of UC01 and UC02
11. Added sequence diagrams for the defined use cases (UC01 and UC02)
12. Draft of the Design Class Diagram from Domain model
13. Added sketches of the UI prototypes
14. Revision control table added to the head of this document

8.3. Iteration 3, review comments and changelog

15. Added external actors to use-case context diagram
16. Added login page diagrams and user interface diagram
17. Added brief format use-case for the *Release Payment* use-case (UC06)
18. Changed Domain Class Diagram format to Contextual Domain Model in UC01
19. Corrected System Sequence Diagram for UC01
20. Changed Domain Class Diagram format to Contextual Domain Model in UC02
21. Corrected System Sequence Diagram for UC02
22. Minor updates to operations contracts
23. Updates/corrections to Interactions Diagrams for the use-cases
24. User interface prototypes changed
25. Created java codes based on the DCD and UC06 release payment.
26. Added state, activity and package diagrams for all documented use case.

8.4. Iteration 4, review comments and changelog

27. Added GRASP principles and refine the design and implementation e.g. Added TaxCalculatorAdaptor & Tax calculator in Design Class Diagram.
28. Based on that GRASP principle updated the Use case, design and class diagram
29. Updated the code accordingly.

8.5. Iteration 5, review comments and changelog

30. Refined the designed patterns
31. Updated the code accordingly.
32. Tried applying design principle