CR Review
Analysis Model

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:
John Virgilio B. Afable Jr.
John Philip O. Chanchico
Lance Andrew B. Lim

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2019-2020

Unique Reference:

The documents are stored in the https://github.com/HappyMonkee/CR-Review/tree/master/02-Requirements%20Engineering/Project%20Deliverables referenced with Group 2 - CR Review - Analysis Model.pdf.

Purpose:

To derive and present the Analysis Model from the Use Case Model and Specifications of the CR Review System.

Audience:

Ms. Rowena Solamo

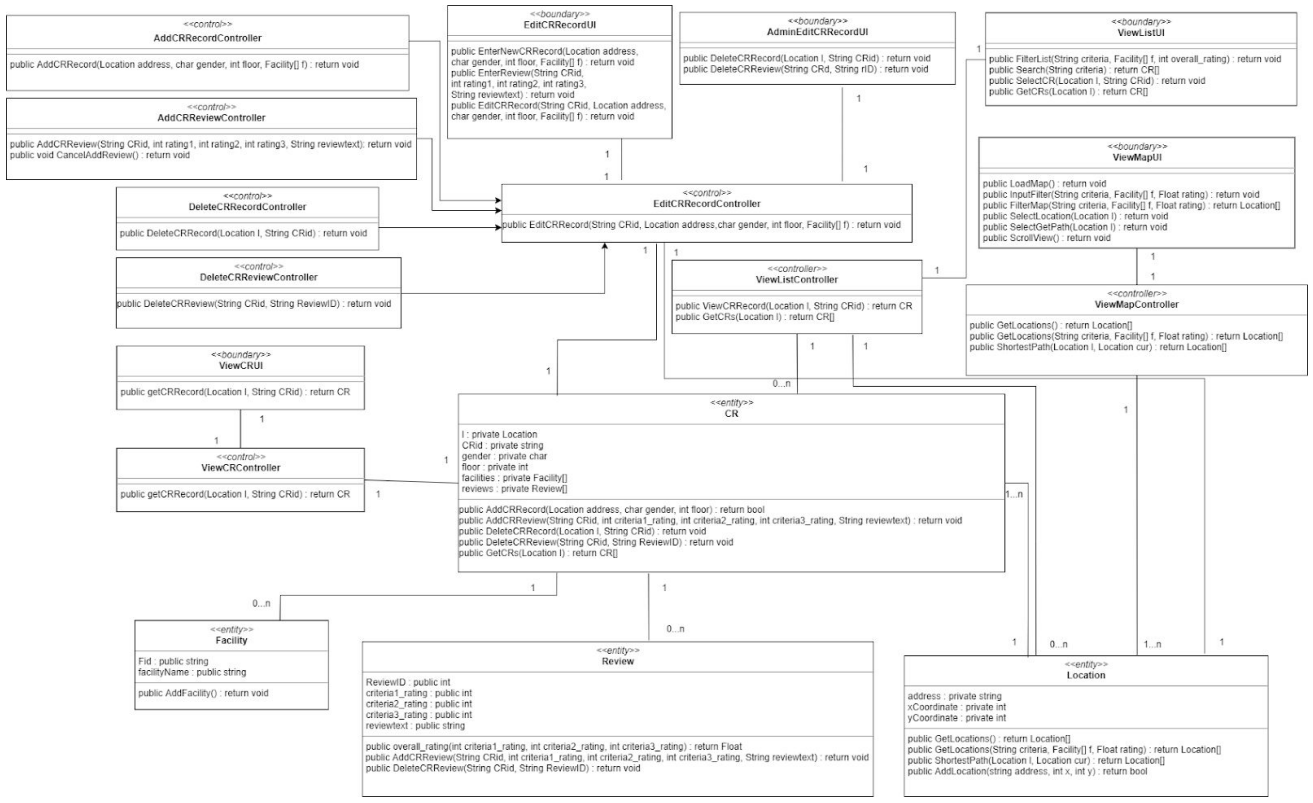
Revision Control:

Revision Date	*Person Responsible*	*Version Number*	*Modification*
09/25/2019	John Philip Ortiz Chanchico	1.0	Initial Document; Added Purpose
09/25/2019	John Philip Ortiz Chanchico	2.0	Added Analysis Model; Added System Name and Audience
09/27/2019	JV Afable	2.1	Modified Analysis Model to include FilterMap() and Search() under ViewMapUI boundary
10/01/2019	Lance Lim	3.0	Updated the Analysis Model, added system description, added descriptions, responsibilities, attributes, and/or functions for boundary, controller, and entity classes.
10/02/2019	Lance Lim	3.1	fixed the descriptions and updated the analysis diagram
10/02/2019	JV Afable	4.0	Added Behavioral Model for View CR Map Use Case
10/02/2019	JV Afable	4.1	Cleaned up Analysis Model, corrected formatting
10/03/2019	Lance Lim	5.0	Added Behavioral Model for Add CR Record Use Case, updated the Analysis Model
10/03/2019	John Philip Ortiz Chanchico	6.0	Added Behavioral Model for Review CR Use Case;
10/03/2019	Lance Lim	6.1	Cleaned up the Analysis Model, updated class descriptions

***System Name*:** CR Record Maintenance System

Description: The system describes the use cases of the CR Record Maintenance System. There are two actors: the User and the Admin. Both the User and Admin may access any functionality related to viewing CRs. User interactions related to editing CR Records are limited to Adding Records and Reviewing CRs while the Admin is limited to Deleting CR Records and Reviews.

***Analysis Model*:**



Boundary Classes:

Class Name	Description
ViewMapUI	This is the interface of both the user and the admin to be able to view the map containing the CRs in UP. *Responsibilities*: public LoadMap() : return void public InputFilter(String criteria, Facility[] f, Float rating) : return void public FilterMap(String criteria, Facility[] f, Float rating) : return Location[] public SelectLocation(Location l) : return void public SelectGetPath(Location l) : return void public ScrollView() : return void
ViewListUI	This is the interface of both the user and admin to be able to view a list of all the CR Records in the database.. *Responsibilities*: public FilterList(String criteria, Facility[] f, int overall_rating) : return void public Search(String criteria) : return CR[] public SelectCR(Location l, String CRid) : return void public GetCRs(Location l) : return CR[]
EditCRRecordUI	This is the interface of both the user and admin, where the user and admin can both: 1) Add CR Records 2) Add CR Reviews 3) Edit CR Records Admins also get to: 1) Delete CR Records 2) Delete CR Reviews *Responsibilities* public EnterNewCRRecord(Location address, char gender, int floor, Facility[] f) : return void public EnterReview(String CRid, int rating1, int rating2, int rating3, String reviewtext) : return void public EditCRRecord(String CRid, Location address,char gender, int floor, Facility[] f) : return void
AdminEditCRRecordUI	This is the interface for the admin, where the admin gets to: 1) Delete CR Records 2) Delete CR Reviews *Responsibilities* public DeleteCRRecord(Location l, String CRid) : return void public DeleteCRReview(String CRd, String rID) : return void
ViewCRUI	This is the interface for both the user and admin to view the CR Record chosen. *Responsibilities*: public getCRRecord(Location l, String CRid) : return CR

Control Classes:

Class Name	Description
EditCRRecordController	This is the control that edits the CR Record in the system. *Responsibilities*: public EditCRRecord(String CRid, Location address,char gender, int floor, Facility[] f) : return void
AddCRRecordController	This is the control that adds a CR Record to the system. It extends EditCRRecordController *Responsibilities*: public AddCRRecord(Location address, char gender, int floor, Facility[] f) : return void
AddCRReviewController	This is the control that adds a review to a CR Record. It extends EditCRRecordController. *Responsibilities*: public AddCRReview(String CRid, int rating1, int rating2, int rating3, String reviewtext): return void public void CancelAddReview() : return void
DeleteCRRecordController	This is the control that deletes a CR Record from the system. It extends EditCRRecordController. *Responsibilities*: public DeleteCRRecord(Location l, String CRid) : return void
DeleteCRReviewController	This is the control that deletes a review from a CR Record. It extends EditCRRecordController. *Responsibilities*: public DeleteCRReview(String CRid, String ReviewID) : return void
ViewCRController	This is the control that gets the details and reviews of a given CR Record to be shown in the interface. *Responsibilities*: public getCRRecord(Location l, String CRid) : return CR
ViewListController	This is the control that gets a filterable list of CR Records in the database. *Responsibilities*: public ViewCRRecord(Location l, String CRid) : return CR public GetCRs(Location l) : return CR[]
ViewMapController	This is the control that gets all the locations of all the CRs in the database to be shown in the map. *Responsibilities*: public GetLocations() : return Location[] public GetLocations(String criteria, Facility[] f, Float rating) : return Location[] public ShortestPath(Location l, Location cur) : return Location[]

Entity Classes:

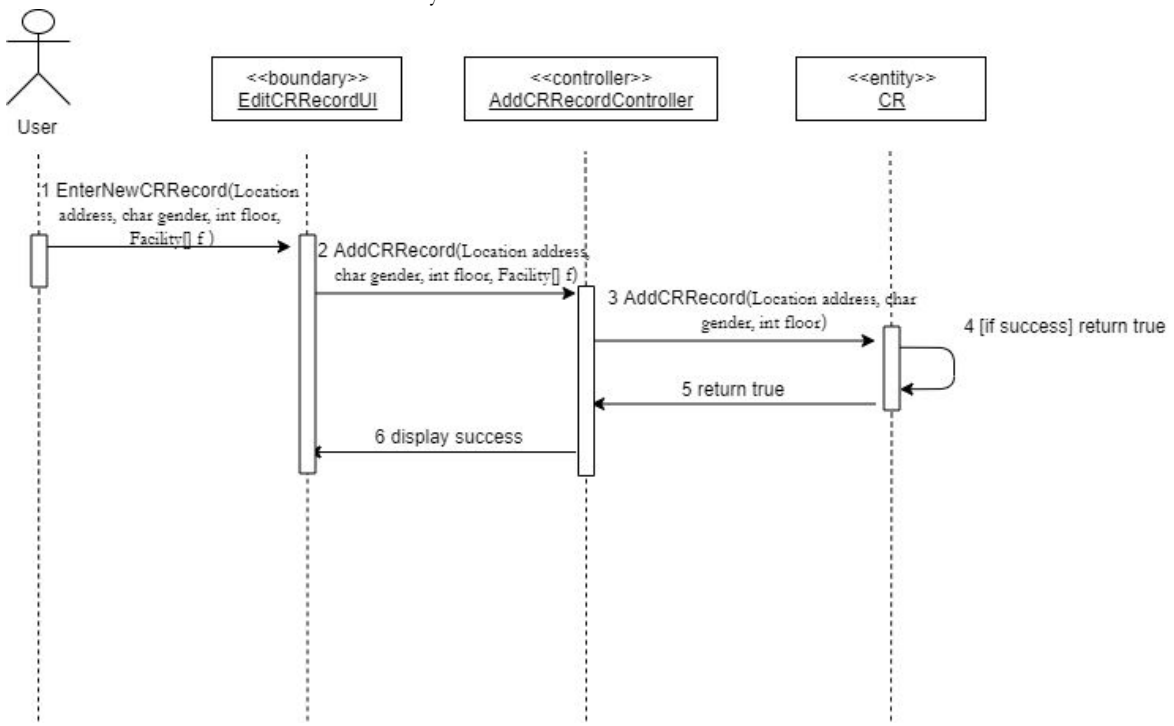
Class Name	Description
CR	This is the entity class CR, which contains the data about the CR Record, as well as all functions associated with the CR record.. *Attributes*: l : private Location CRid : private string gender : private char floor : private int facilities : private Facility[] reviews : private Review[] *Functions*: public AddCRRecord(Location address, char gender, int floor) : return bool public AddCRReview(String CRid, int criteria1_rating, int criteria2_rating, int criteria3_rating, String reviewtext) : return void public DeleteCRRecord(Location l, String CRid) : return void public DeleteCRReview(String CRid, String ReviewID) : return void public GetCRs(Location l) : return CR[]
Location	This is the entity class Location, which contains the data about the locations of the CRs. *Attributes*: address : private string xCoordinate : private int yCoordinate : private int *Functions*: public GetLocations() : return Location[] public GetLocations(String criteria, Facility[] f, Float rating) : return Location[] public ShortestPath(Location l, Location cur) : return Location[] public AddLocation(string address, int x, int y) : return bool
Review	This is the entity class Review, which contains the data about the reviews of each CR Record. *Attributes*: ReviewID : public int criteria1_rating : public int criteria2_rating : public int criteria3_rating : public int reviewtext : public string *Functions*: public overall_rating(int criteria1_rating, int criteria2_rating, int criteria3_rating) : return Float public AddCRReview(String CRid, int criteria1_rating, int criteria2_rating, int criteria3_rating, String reviewtext) : return void public DeleteCRReview(String CRid, String ReviewID) : return void
Facility	This is the entity class Facility, which contains the data about the facilities of every CR record. *Attributes*: Fid : public string facilityName : public string *Functions* public AddFacility() : return void

Behavioral Model:

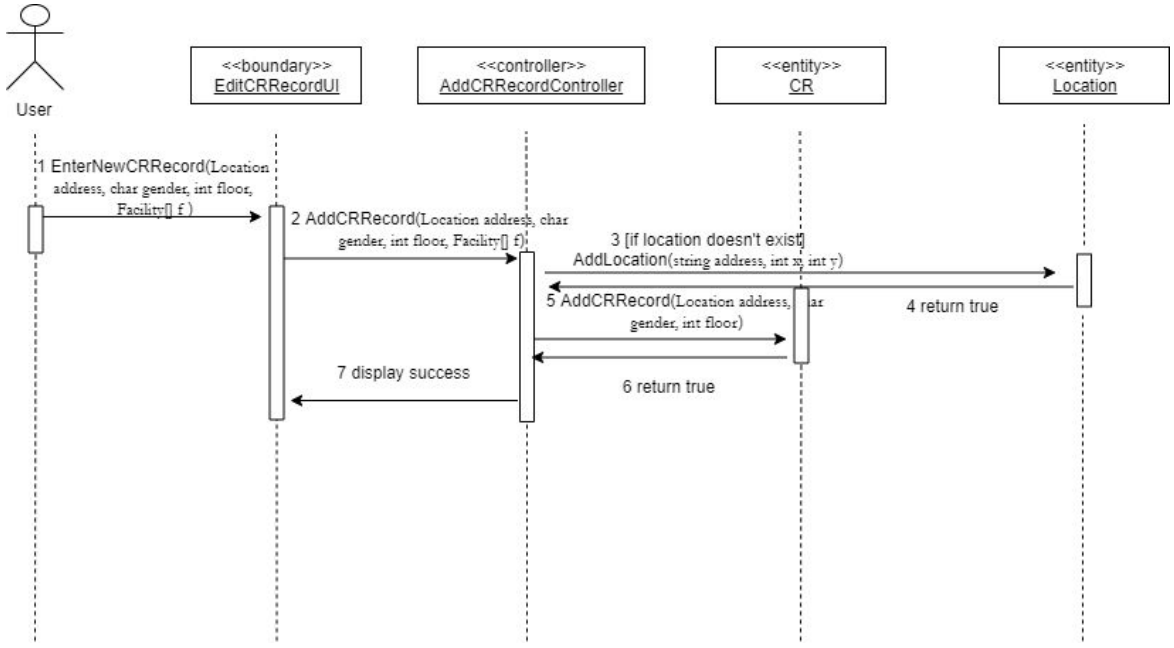
***Use-Case Name*:** 1.1 Add CR Record

Description: Accessible only to the User. Creates a new CR Record by entering the details for location, available supplies and functional facilities. Details will be uploaded to the database and the newly made CR Record will be available for editing, reviewing, and deleting.

Scenario 1: Added CR Record successfully

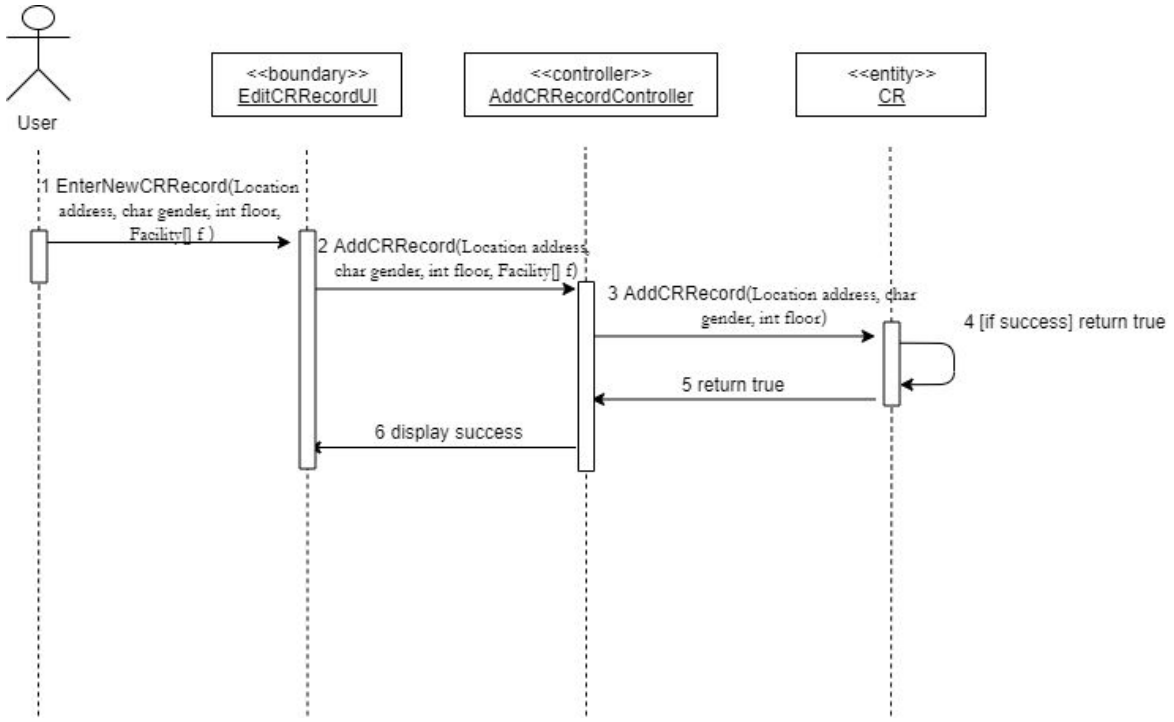


Scenario 2: CR Location added does not exist in the map.

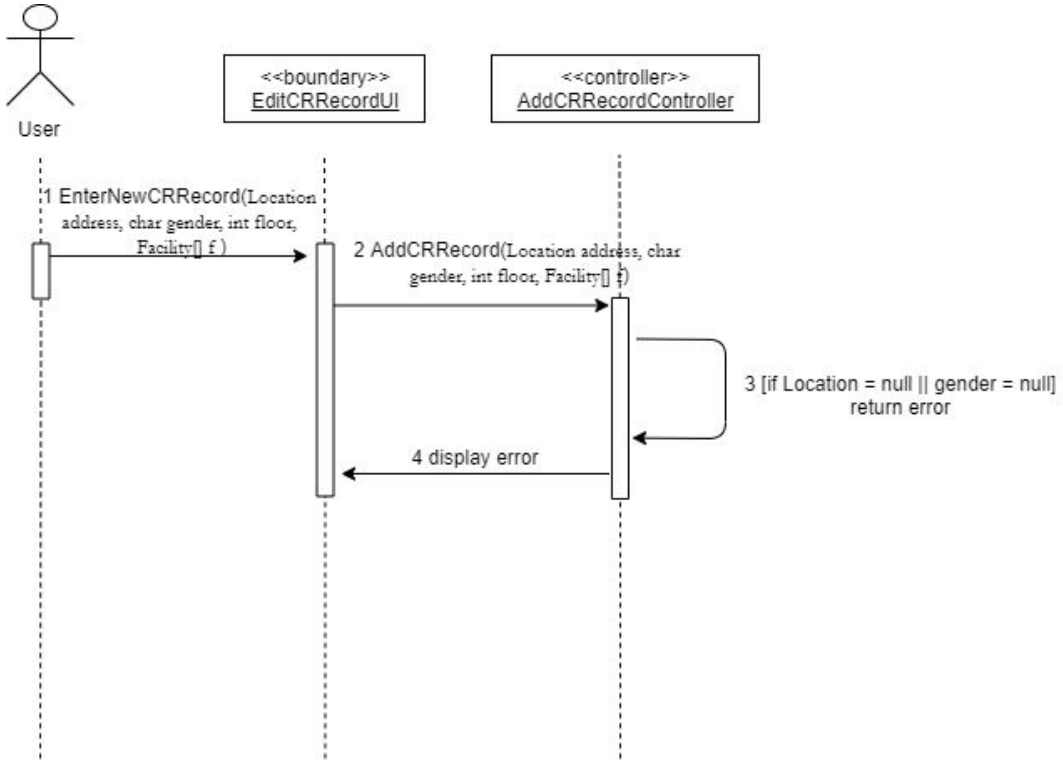


Scenario 3: Add CR Record with no available Facilities

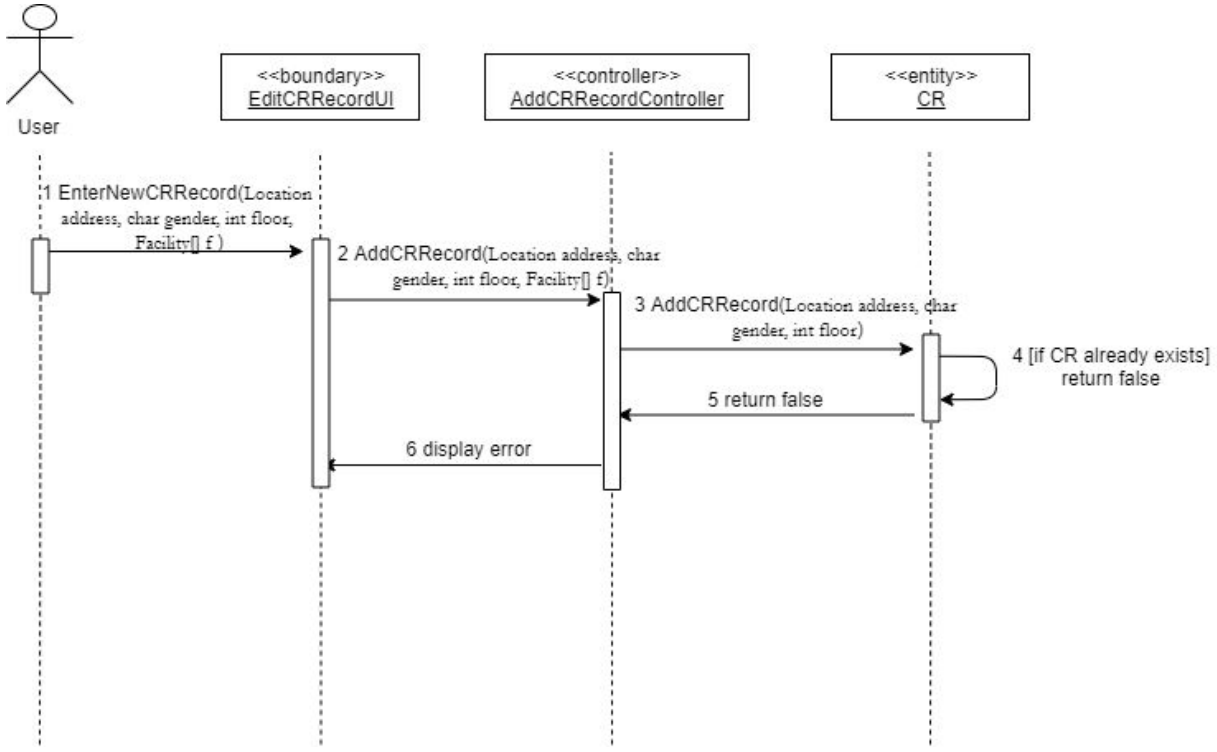
It is possible for the Facilities of a CR entity to be null, (or an empty array), so the add will still be successful.



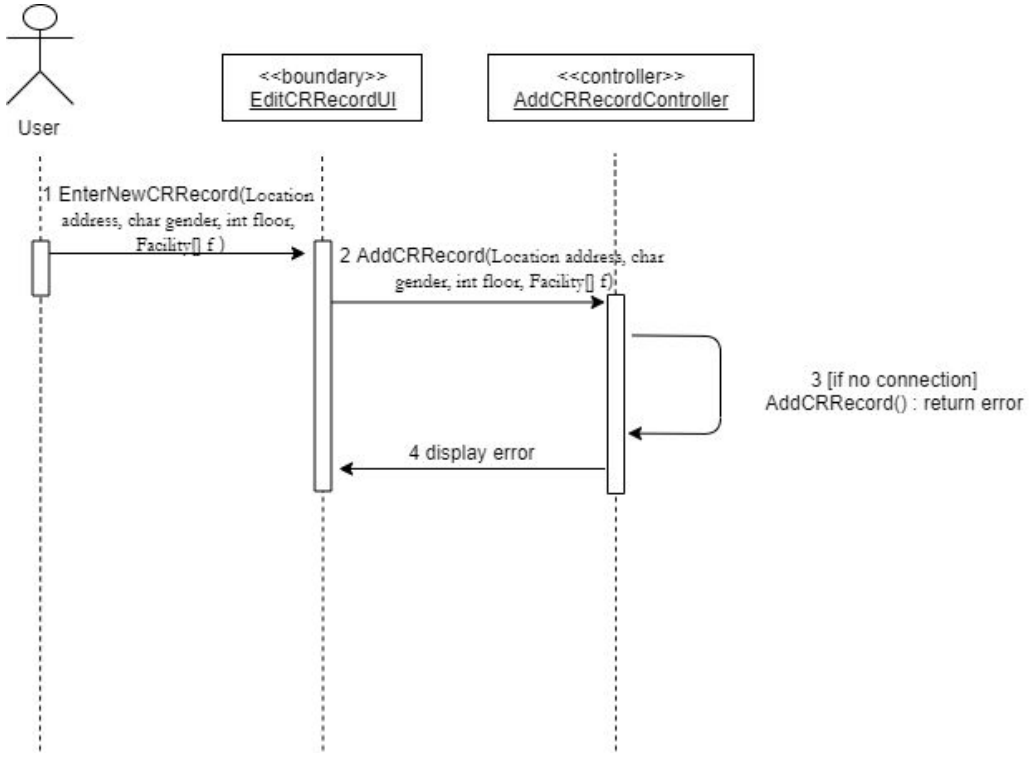
Scenario 4: No input for location and/or gender



Scenario 5: CR Record already exists (Similar record exists).



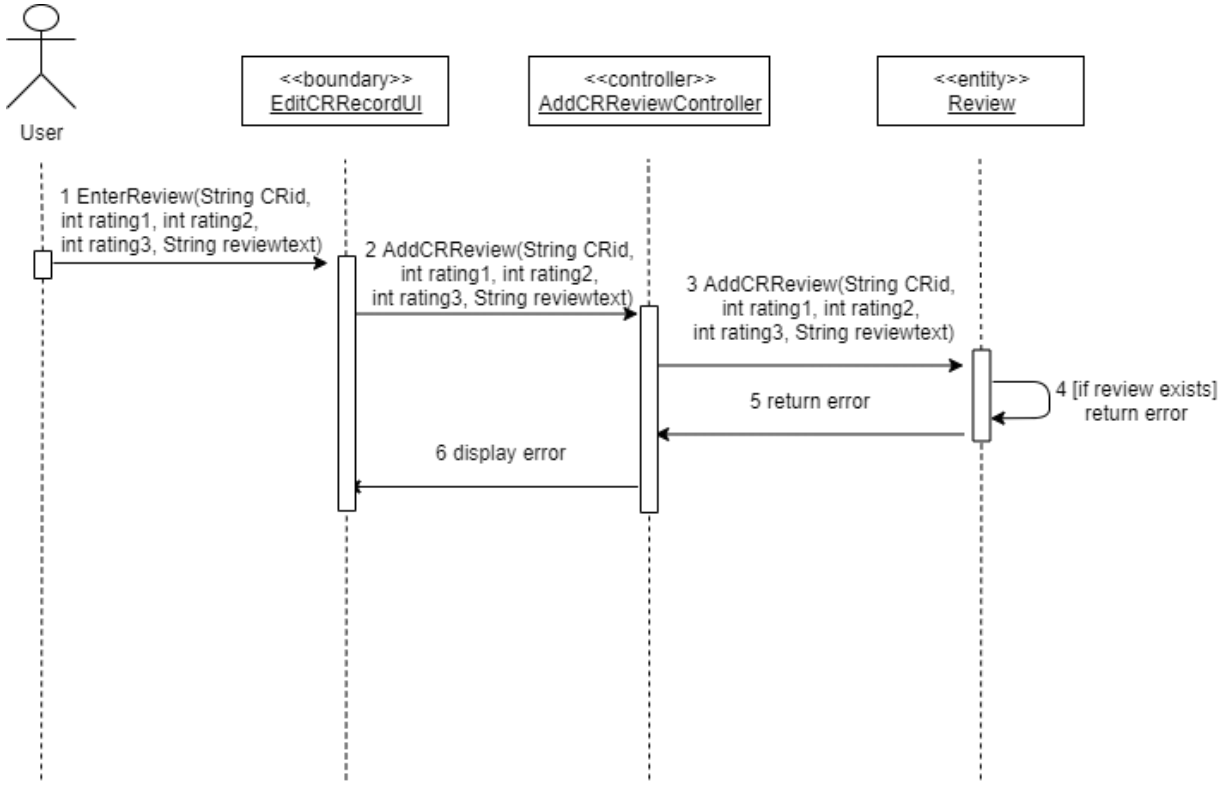
Scenario 6: No database connection



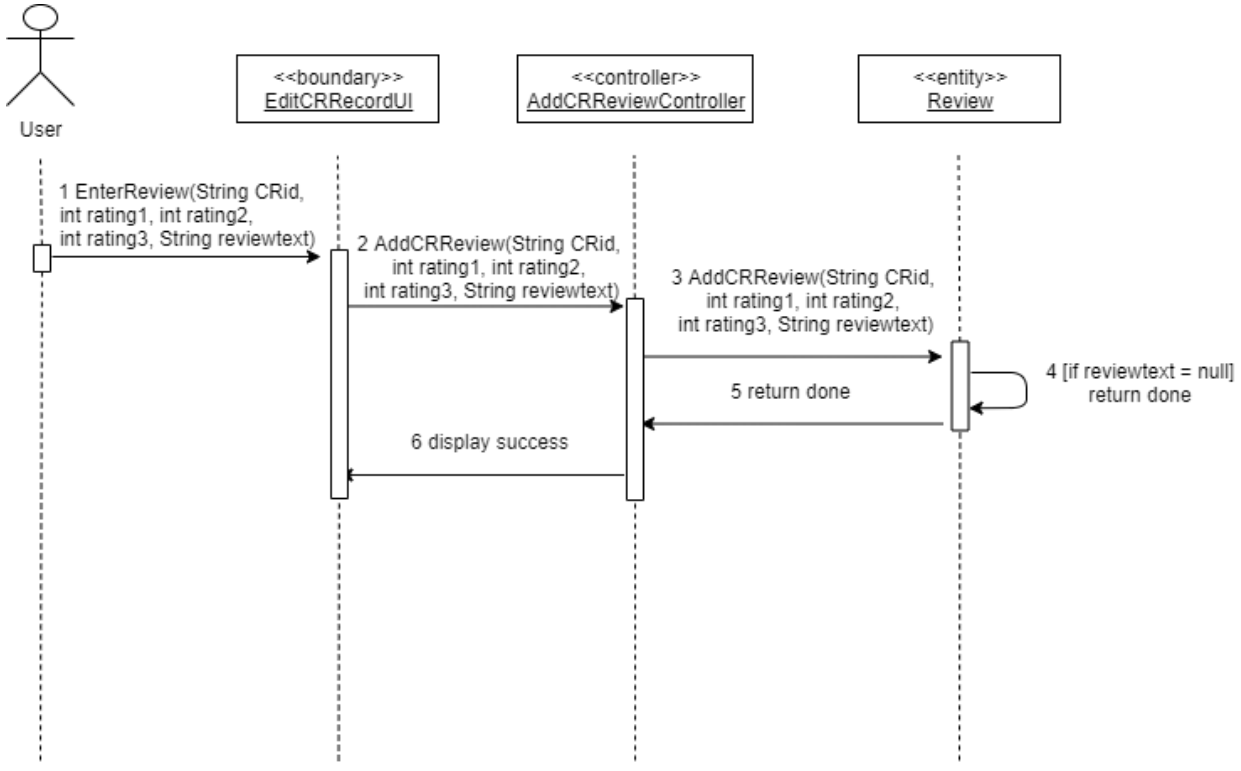
***Use-Case Name*:** 1.2 Review CR

Description: Enter text/string entry into a field. Afterwards, ratings of various criteria are asked. Average rating of these criteria is used as the overall rating of the CR. Entry is inserted under the corresponding CR reviewed.

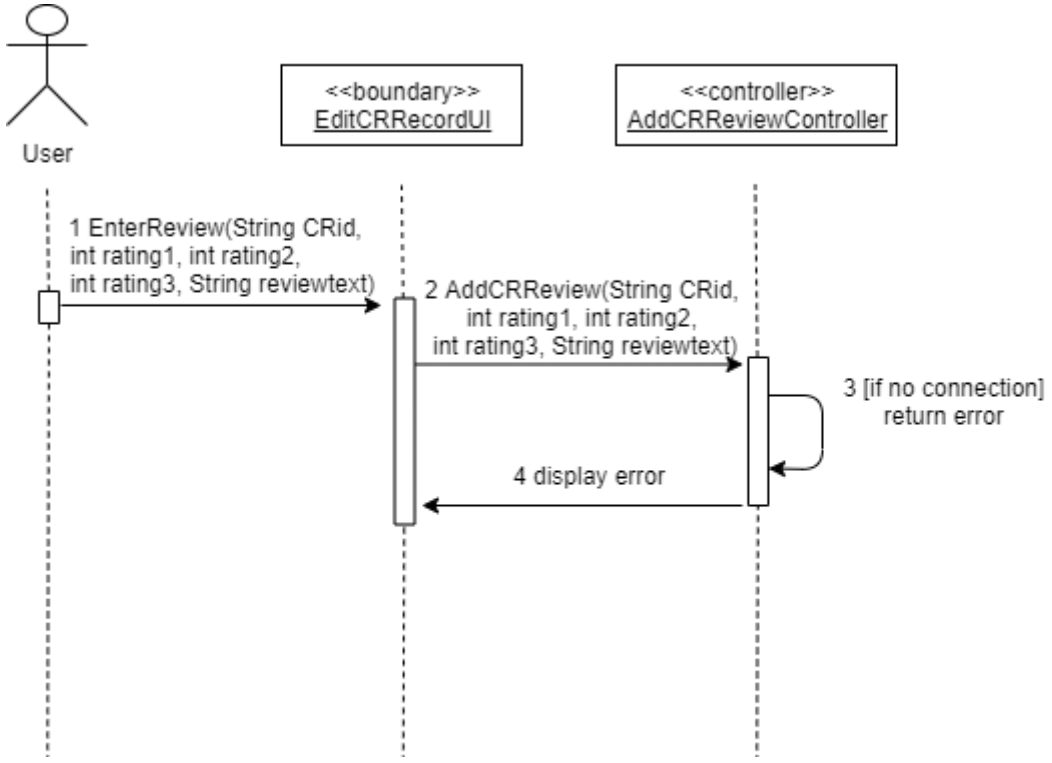
Scenario 1: Similar review exists



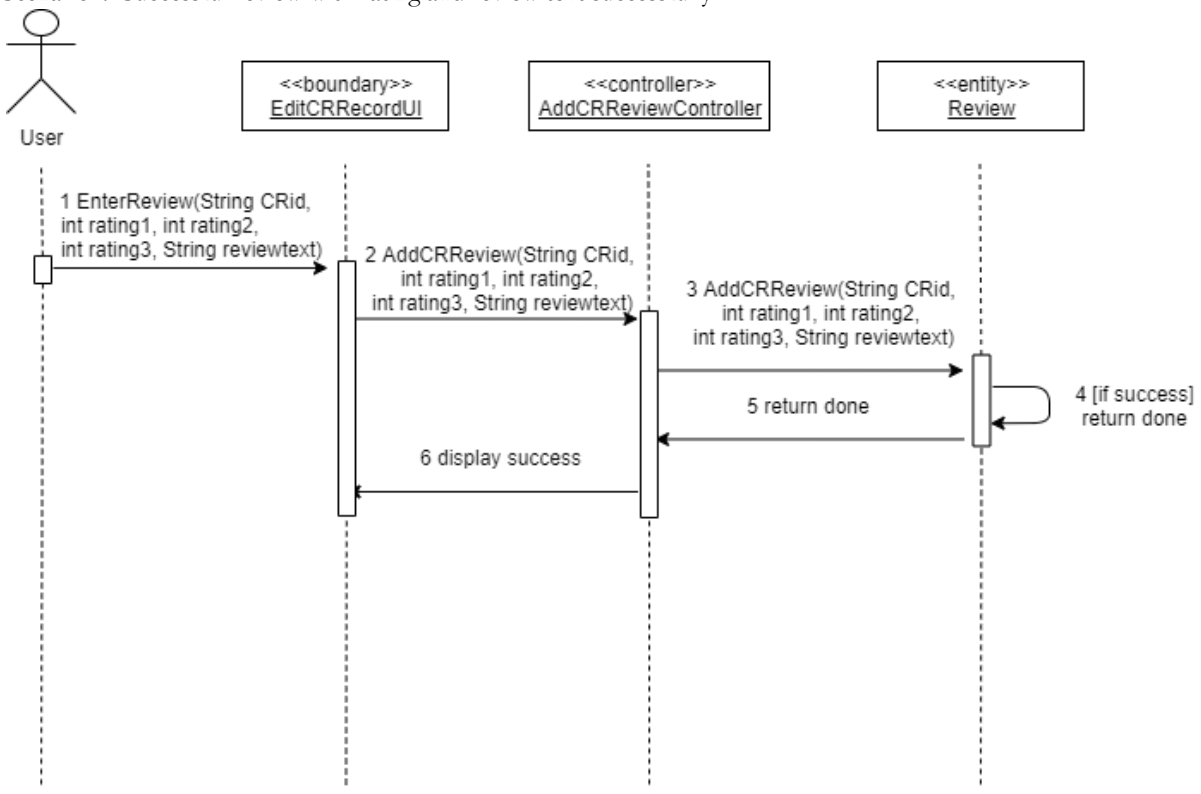
Scenario 2: No text is entered as review



Scenario 3: No database connection



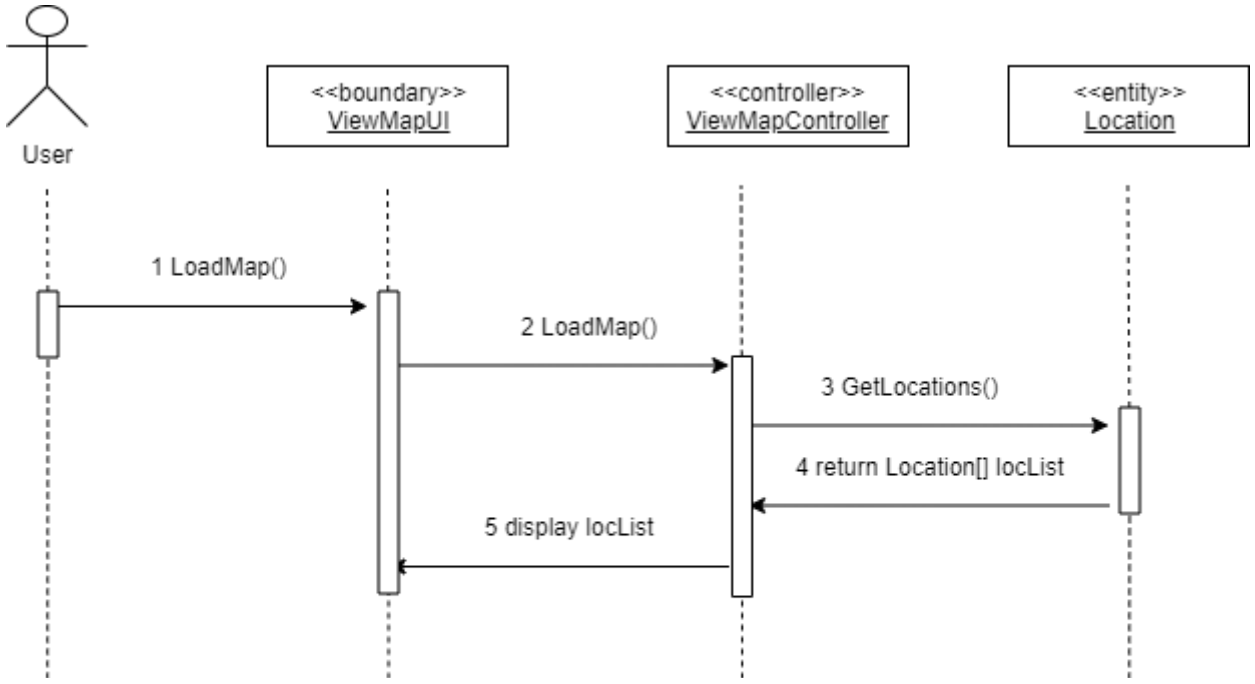
Scenario 4: Successful review with rating and review text successfully



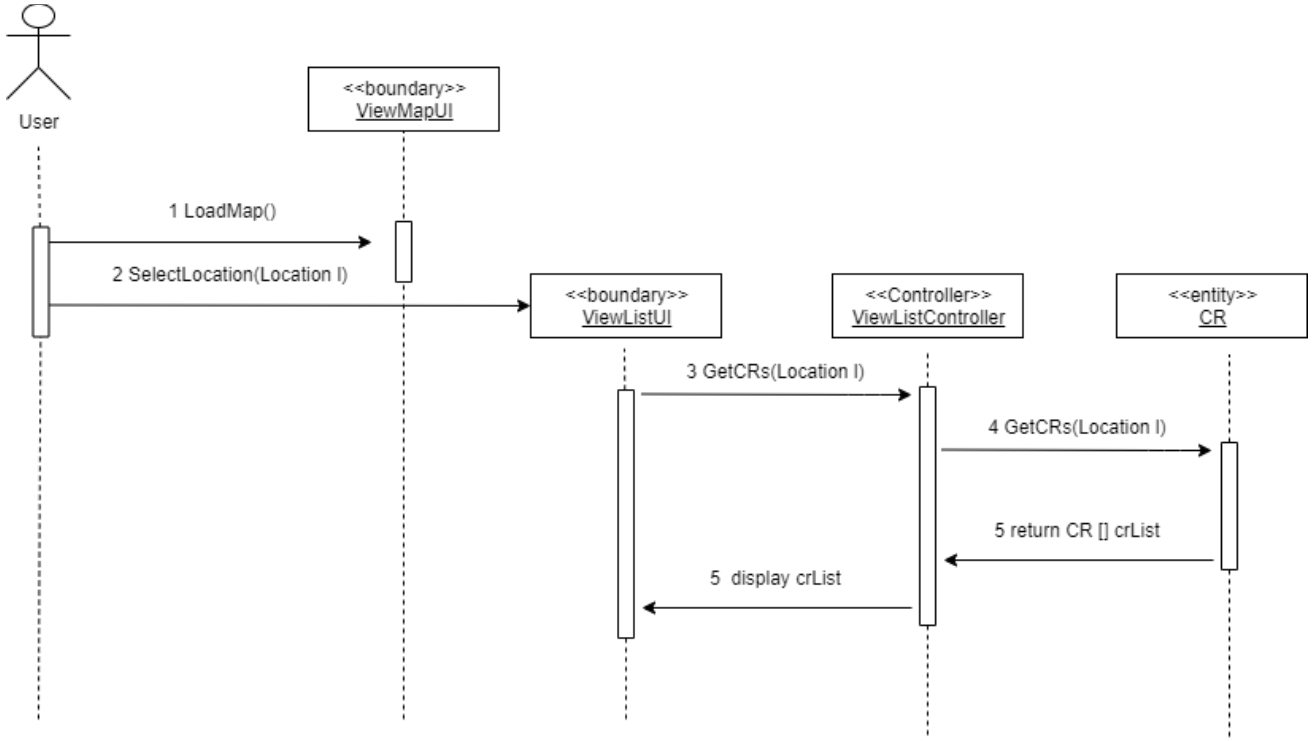
***Use-Case Name*:** 4.0 View CR Map

Description: Opens a map of UP Diliman. This map contains location markers of all the CRs in the current CR list. Location markers of CRs can be accessed to view the details of said CR

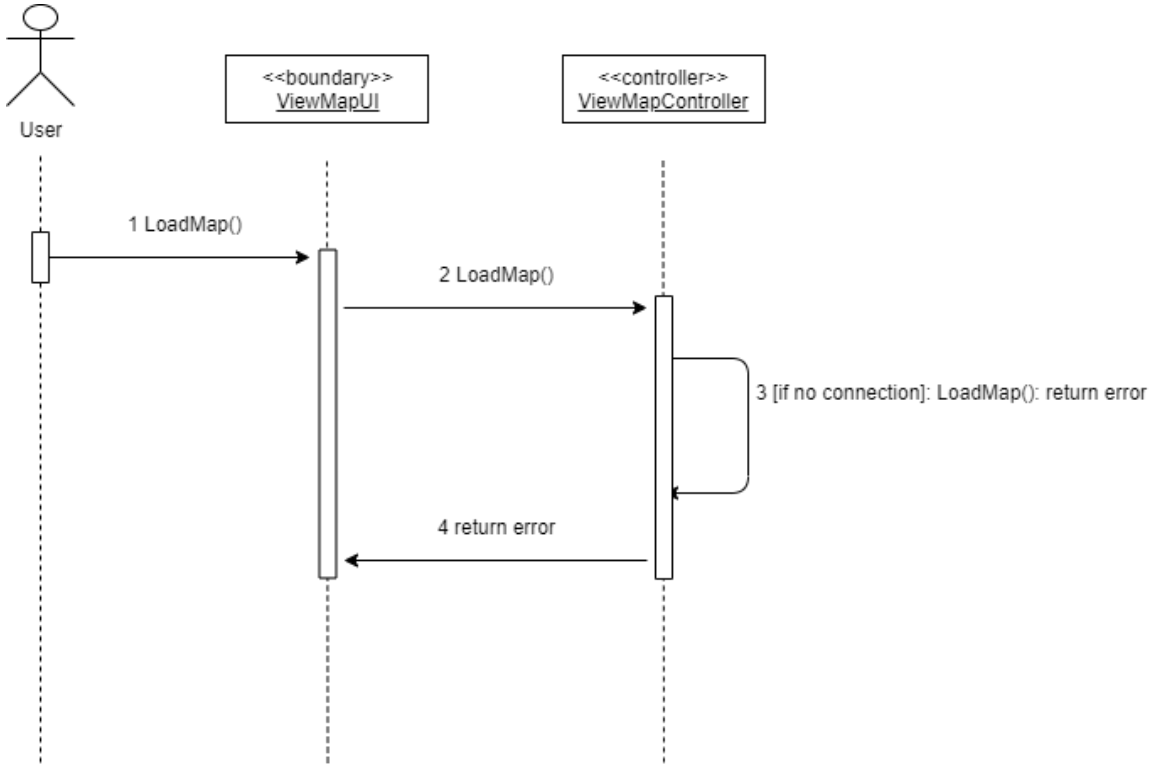
Scenario 1: Basic Flow



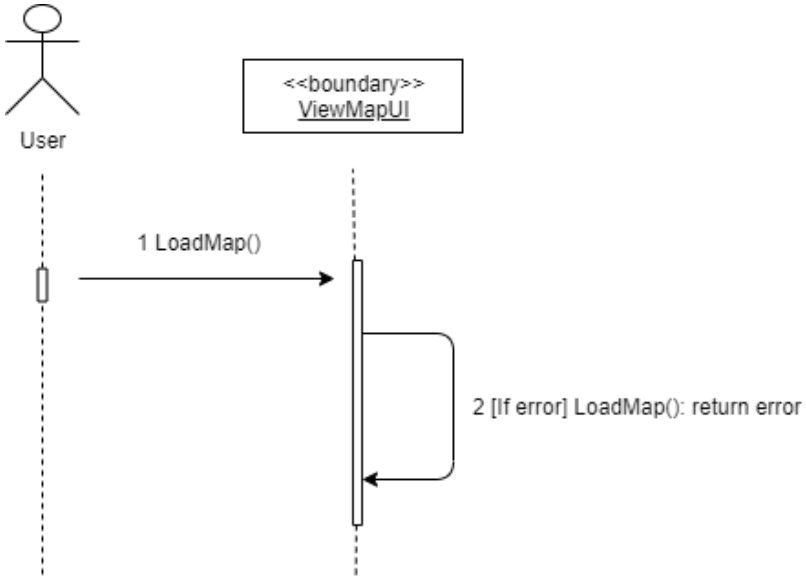
Scenario 2: Select Location



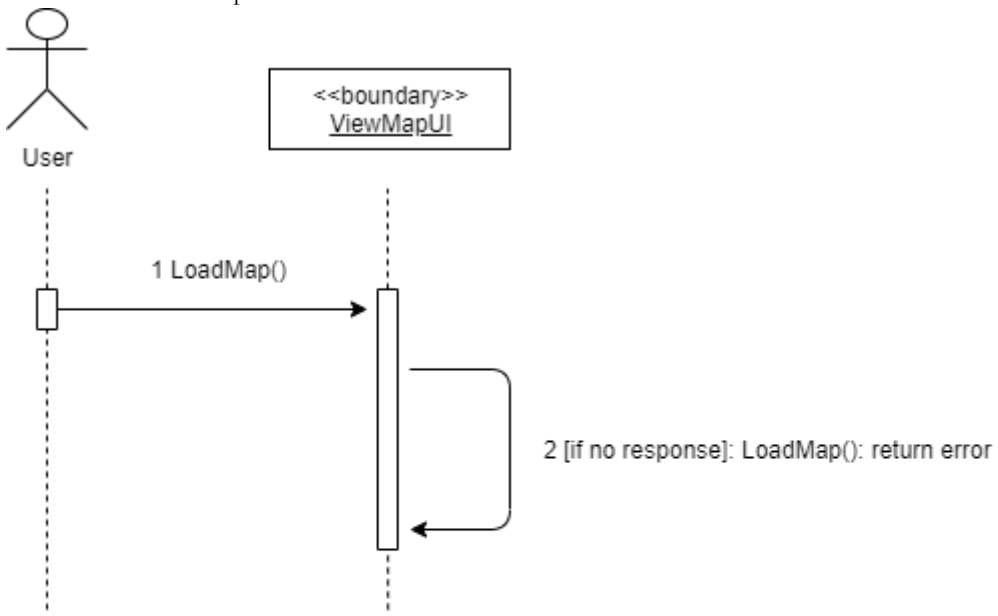
Scenario 3: No Database Connection



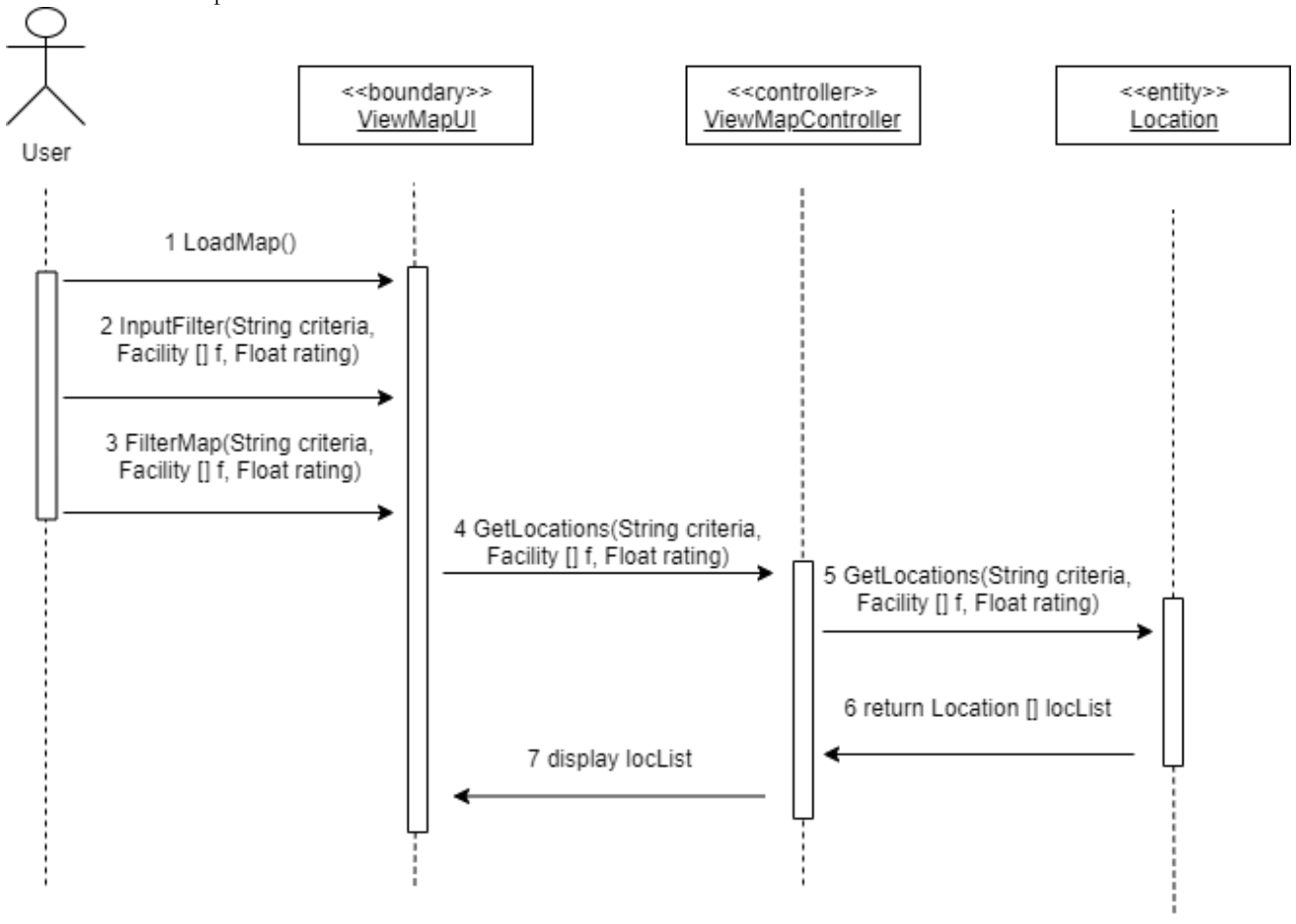
Scenario 4: Load Map Error



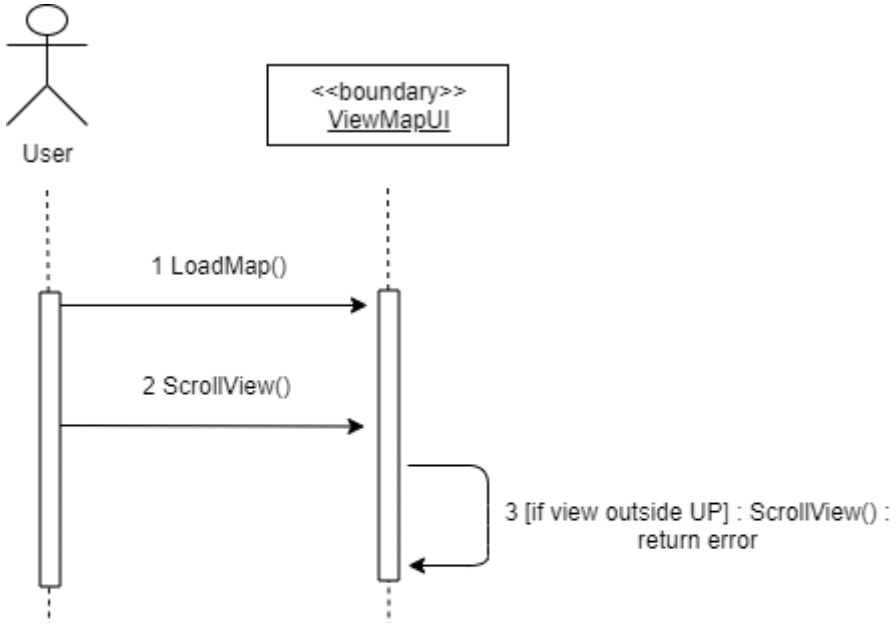
Scenario 5: API Unresponsive



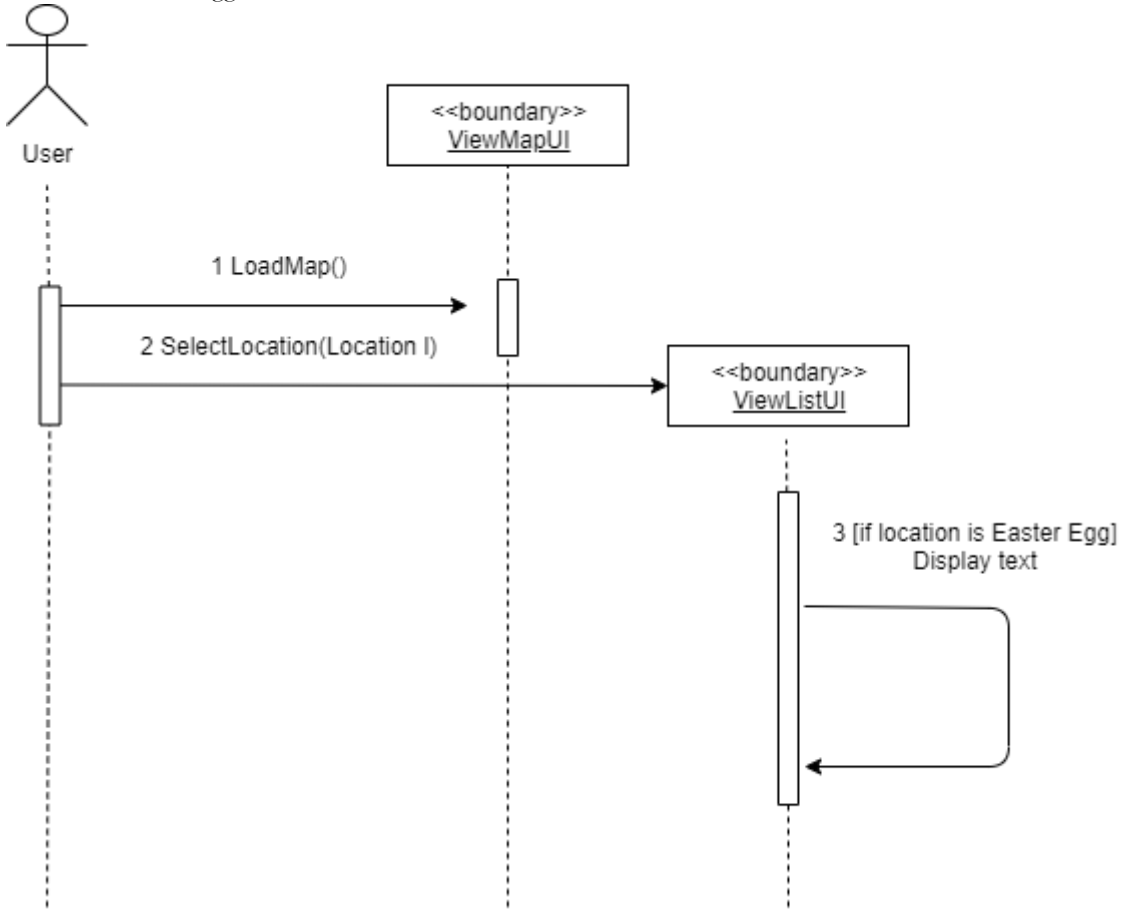
Scenario 6: Filter Map



Scenario 7: View Outside UP



Scenario 8: Easter Egg



Scenario 9: Get Shortest Path