Study Space
Analysis Document
Version 1.3

1. Introduction

Study Space is a web-based application that allows students in any university to post and find study group locations. By doing so, students are able to help each other achieve their academic goals as well as make new acquaintances.

2. Object Classification

2.1 Entity Objects

Beacon- A study location with details of the location created by a RegisteredUser for other Users and RegisteredUsers to join.
BeaconList- A list of Beacons.
ChatDB- A way for all GuestUsers and RegisteredUsers to communicate through text.
GuestUser- A Human who does not own an account and is using the application.
UserDB- A database to store RegisteredUsers credentials.
Profile- A RegisteredUser's profile they can edit.
RegisteredUser- A GuestUser that created an account with username and password.

2.2 Boundary Objects

BeaconChatWindow- A Chat including all GuestUsers and RegisteredUsers in that Beacon.
BeaconConfirmationWindow- A window confirming the credentials of a Beacon.
BeaconControlWindow- A window with all options to control the Beacon.
BeaconCreationWindow- Where a RegisteredUser can create a Beacon by entering location and course code then can control that Beacon.
BeaconFailedJoinWindow- A window shown when joining Beacon has failed.
BeaconJoinedWindow- A window notifying a GuestUser or RegisteredUser that they joined a Beacon.
ChatBar- A bar containing all chat instances to that GuestUser or RegisteredUser.
ChatWindow- A window of which a GuestUser or RegisteredUser sends and receives messages.
HomeWindow- The main window view of all operations to GuestUsers and RegisteredUsers.
LoginWindow- A window where GuestUsers can enter their credentials to login to their RegisteredUser account.
ProfileWindow- Displays a RegisteredUsers Profile.
SearchWindow- A window where GuestUsers and RegisteredUsers apply filters to BeaconList.

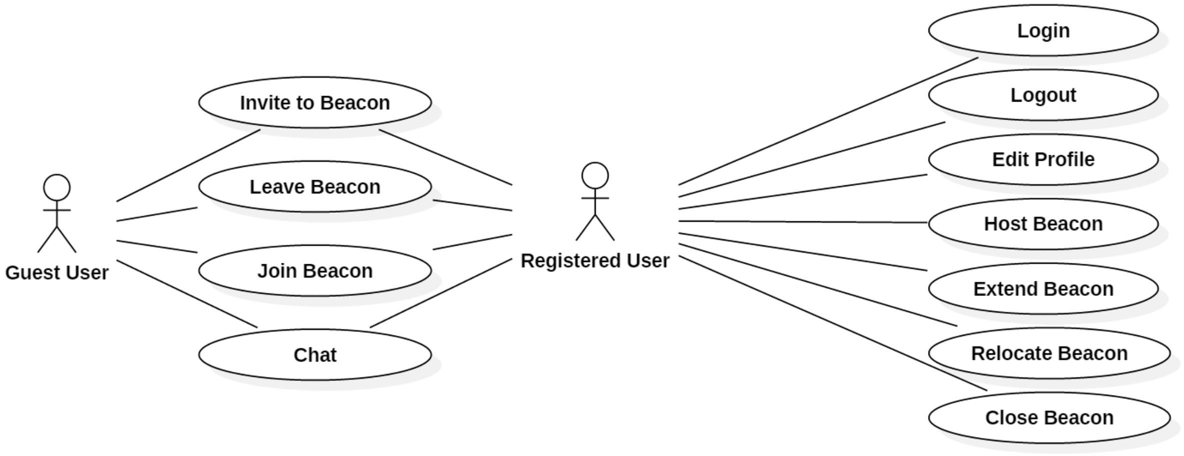
2.3 Control Objects

ApplySearchFilters- Applies filters to BeaconList.
BackToSearch- Redirects GuestUser or RegisteredUser back to SearchWindow after a failed search.
CreateBeacon- Creates the beacon.
CloseBeacon- Allows RegisteredUser controlling the Beacon to close the Beacon.
CancelChanges- Cancels changes made to a RegisteredUsers ProfileWindow.
DisplayResults- Displays results of a search.
EditProfile- Applies changes to a RegisteredUsers Profile.

ExtendBeacon- Allows RegisteredUser controlling the Beacon to extend the time of their Beacon.
EnterInfo- Writes RegisteredUser entered info to the Beacon.
FailedCreation- Notifies the RegisteredUser controlling the Beacon of failed Beacon creation.
InviteUser- Invites another GuestUser or RegisteredUser to chat.
JoinBeacon- Joins a GuestUser or RegisteredUser to a Beacon.
JoinChat- Controller to join a GuestUser or RegisteredUser to a chat.
LeaveBeacon- Removes a GuestUser or RegisteredUser from a Beacon they are in.
Login- Logs in a GuestUser to their RegisteredUser account.
Logout- Logs out a RegisteredUser from their account.
LeaveChat- Controller to GuestUser or RegisteredUser leaves Chat.
NotifyUser- Notifies GuestUser or RegisteredUser if an invitation to a Beacon.
SaveChanges- Saves changes made to a RegisteredUser ProfileWindow.
SendMsg- Controls sending a message to a Chat.
RelocateBeacon- Allows RegisteredUser controlling the Beacon to relocate their Beacon.
RegisterBeacon- Registers Beacon to BeaconList.
ValidateBeacon- Validates Beacon creation.
ValidateUser- Validates a RegisteredUsers credentials upon login.

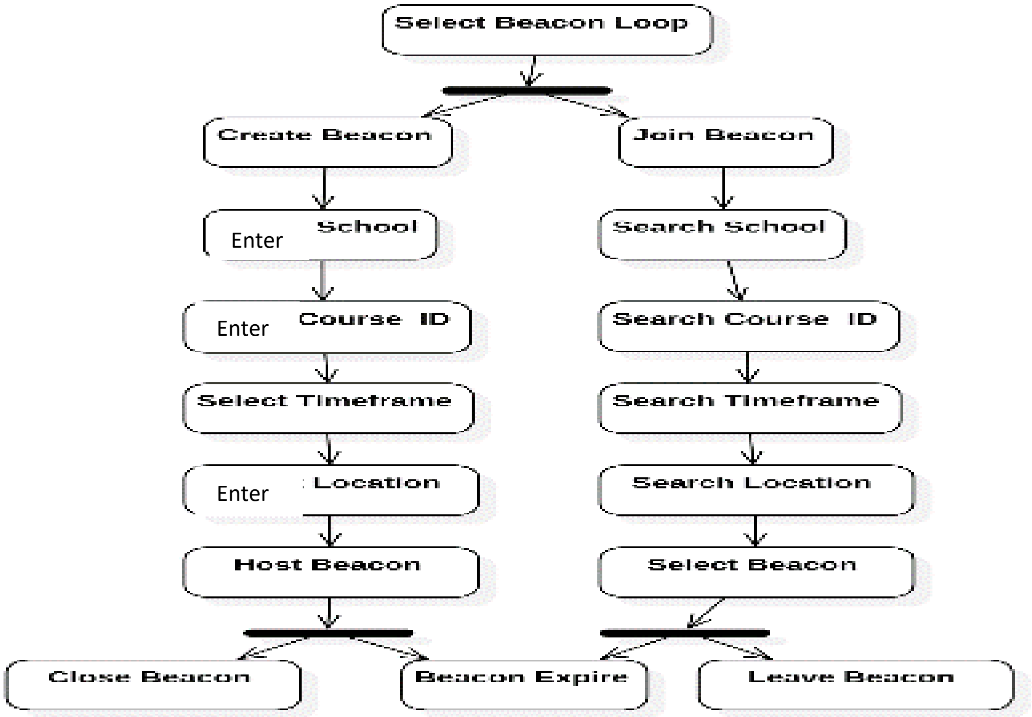
3.Diagrams

Use Case

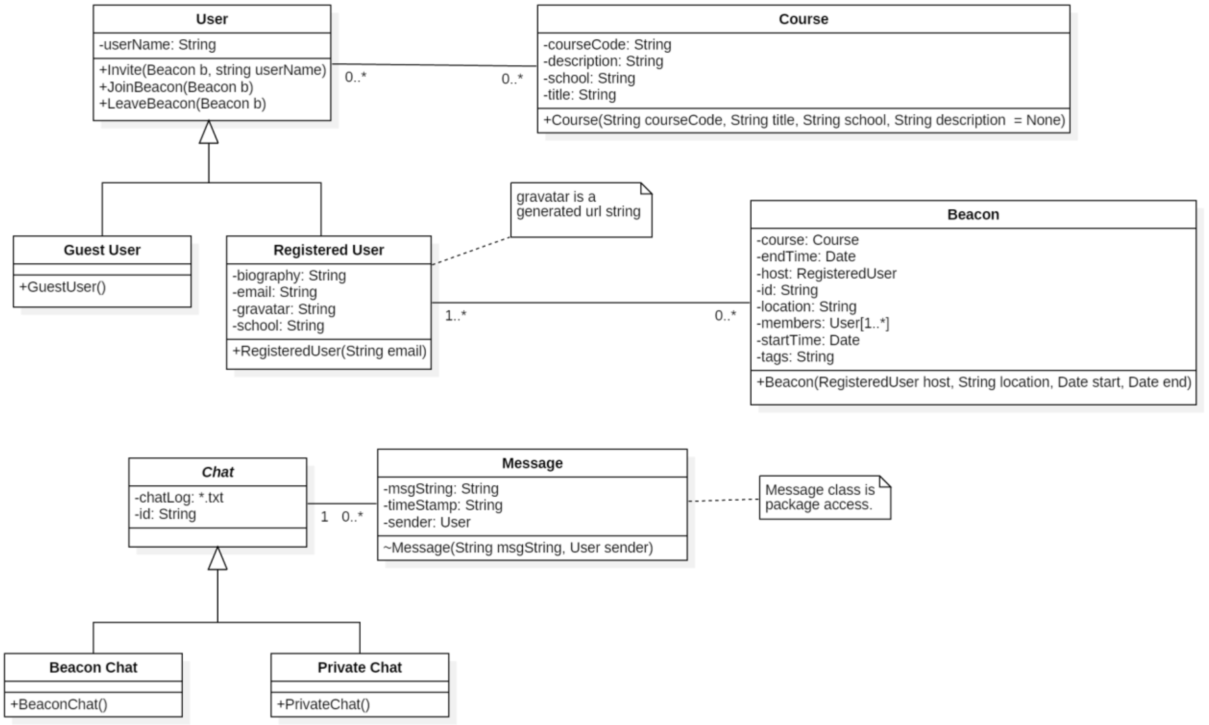


State Diagram

Model::StateMachine1::ping select loop



Class Diagram



Class Diagram Description

This class diagrams outlines each class to be used within the web application. A brief description of each class is given below:

Beacon: Each Beacon is associated with 0 or more Beacon(s), while each Beacon is associated with 1 or more RegisteredUser(s). The Beacon class stores the location, start and end time of the Beacon, as well as other information pertaining to that particular Beacon's RegisteredUser and Course.

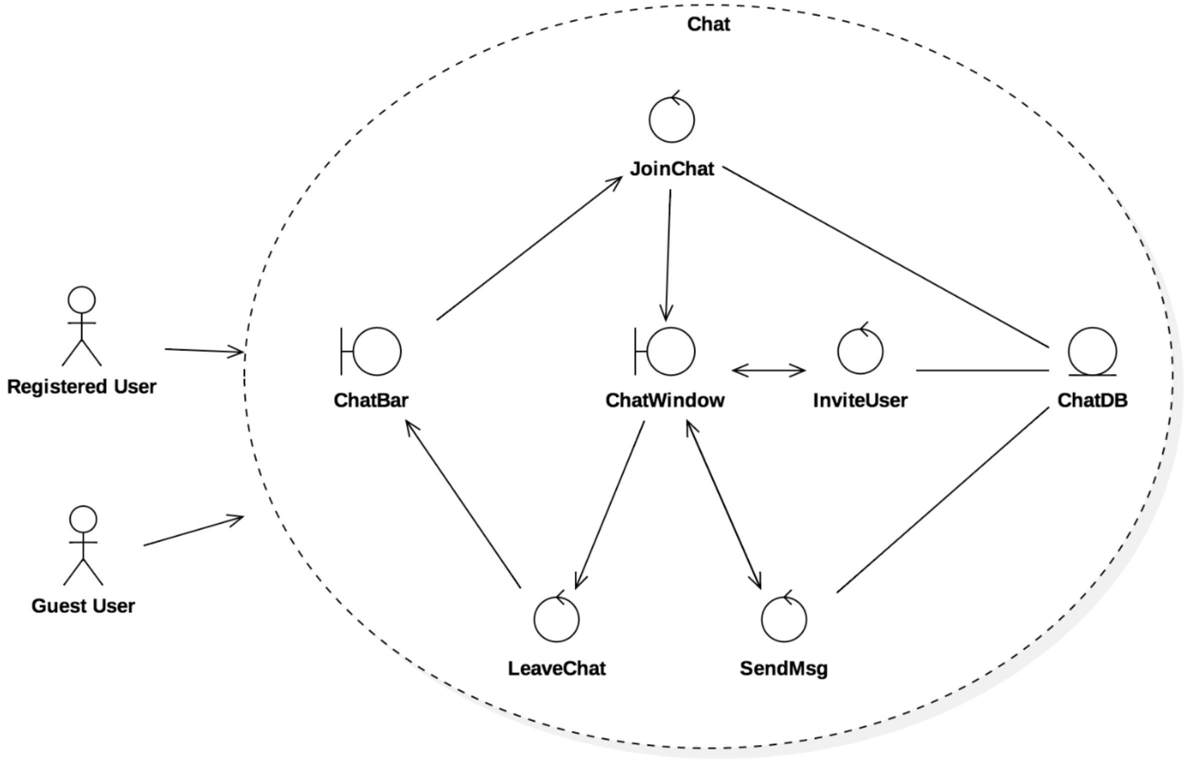
Course: Each GuestUser or RegisteredUser may be associated with 0 or more Course(s). The Course class stores user-generated information pertaining to each course the User identifies as enrolled in.

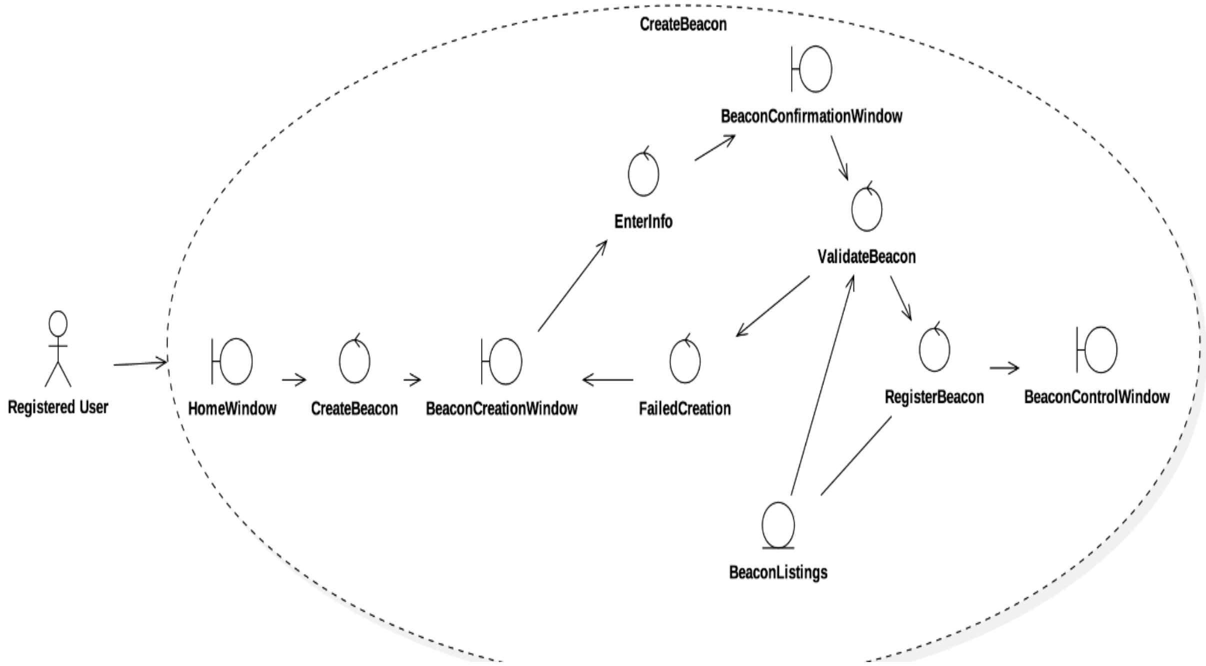
Chat: The Chat class operates as an abstract class which implements two child classes – BeaconChat and PrivateChat. Each chat is associated with 0 or more Message(s). The BeaconChat class allows for communication between members of a Beacon. This class requires a separate ID generation process from that implemented by PrivateChat given the fact that PrivateChat is only between two people.

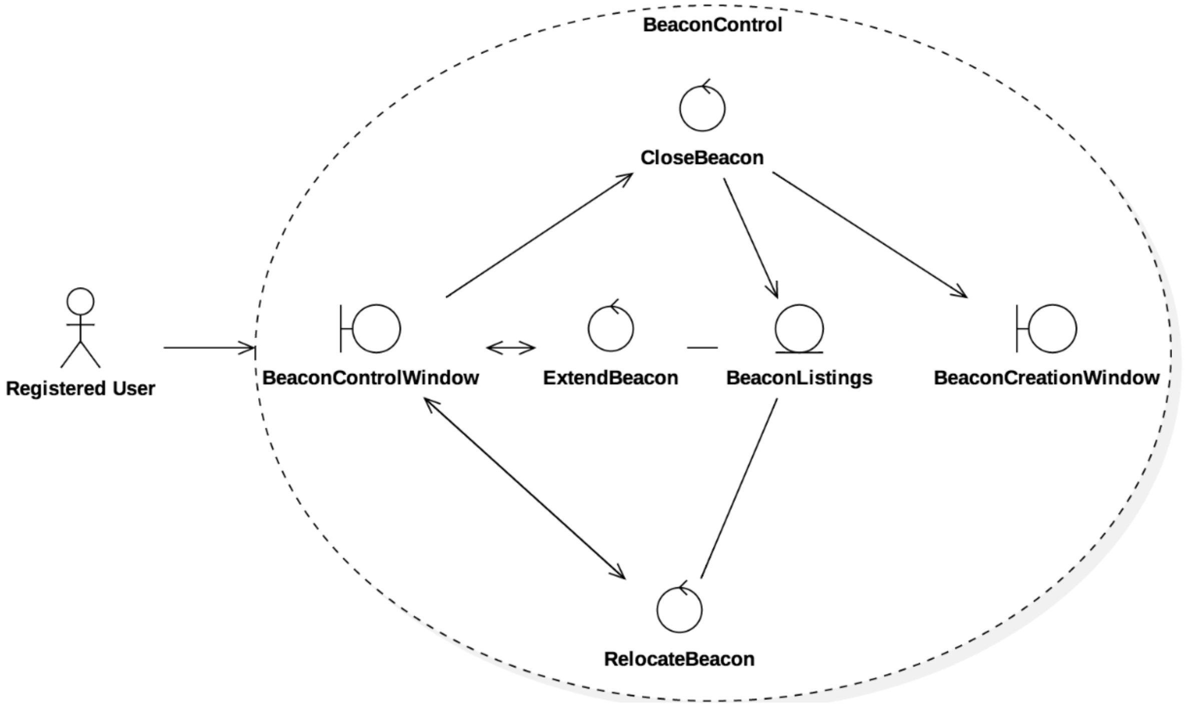
Message: The Message class is associated with 1 and only 1 chat. Each Message is package access and may only be accessed from within itself or through the Chat class. Each Message will be associated with a User, a text string, as well as a timestamp. This timestamp is not included in the Message constructer and will be identified at runtime by the CPU.

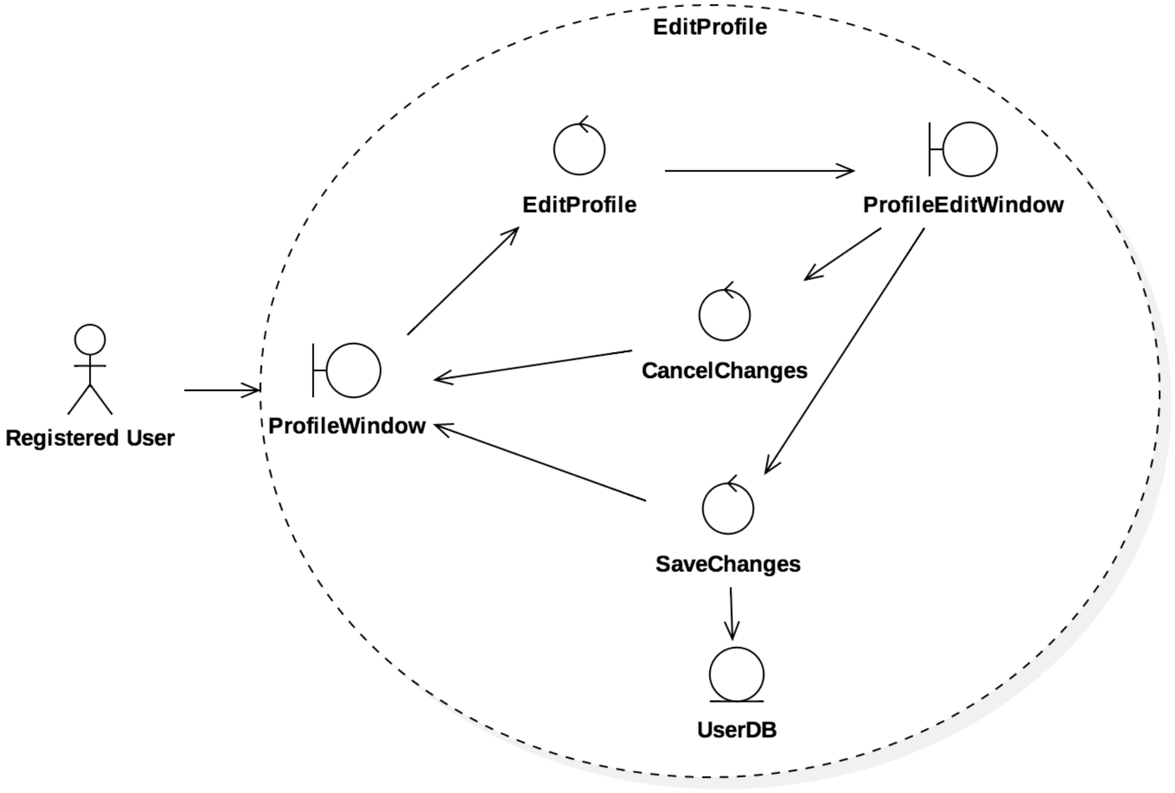
User: The User class is the parent class of the GuestUser and the RegisteredUser. While both GuestUser and RegisteredUser inherit methods from the parent User class, only a Registered User may create a Beacon. The RegisteredUser class also stores that user's Gravitar (Globally Recognized Avatar), which associates a picture with each RegisteredUser.

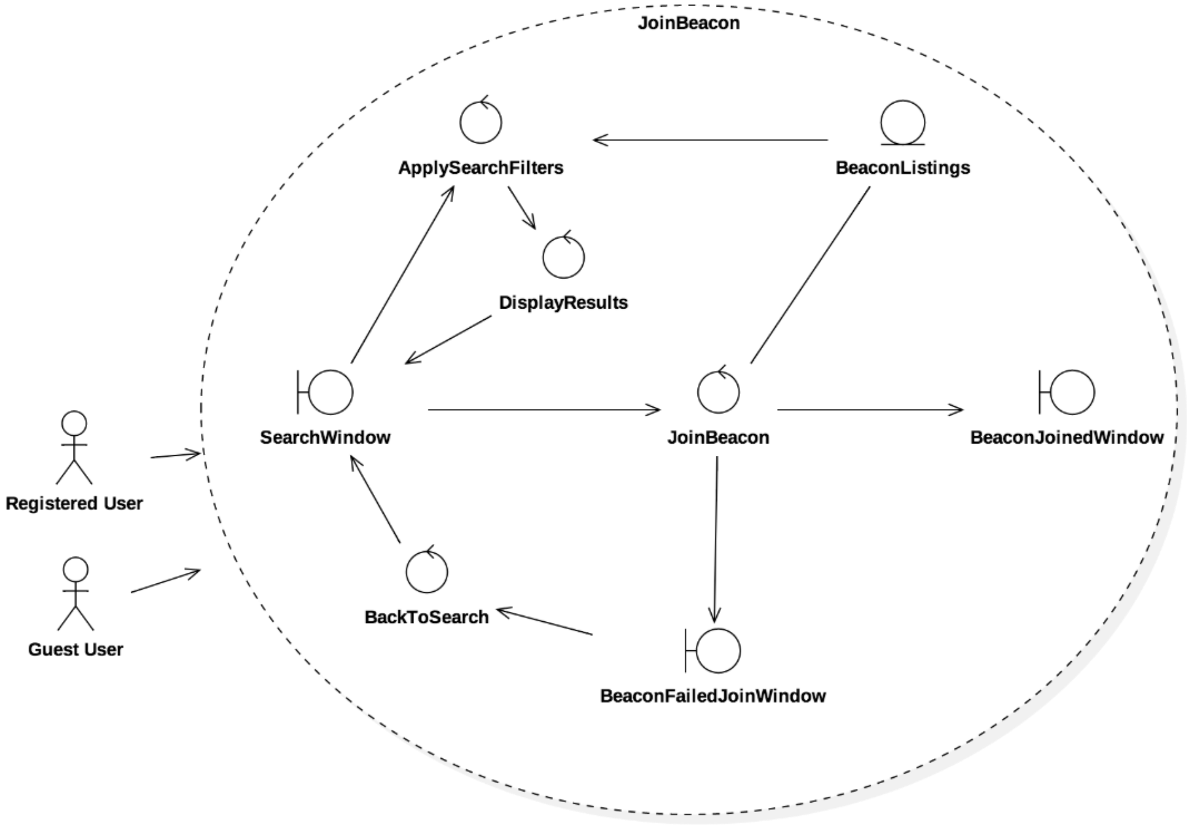
Robustness Diagrams

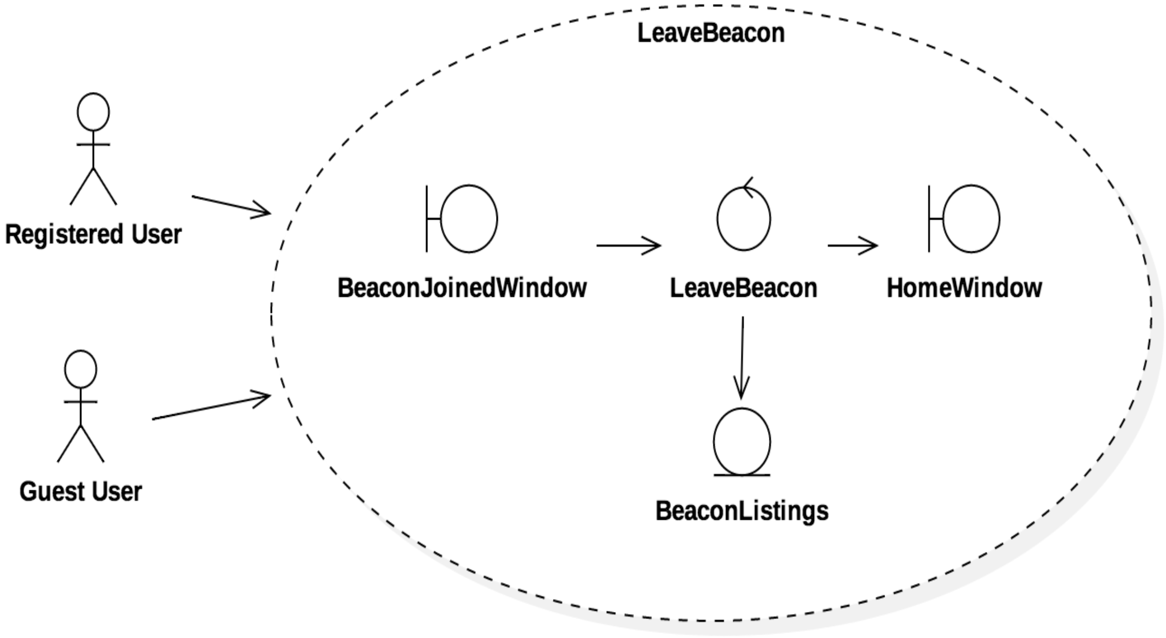


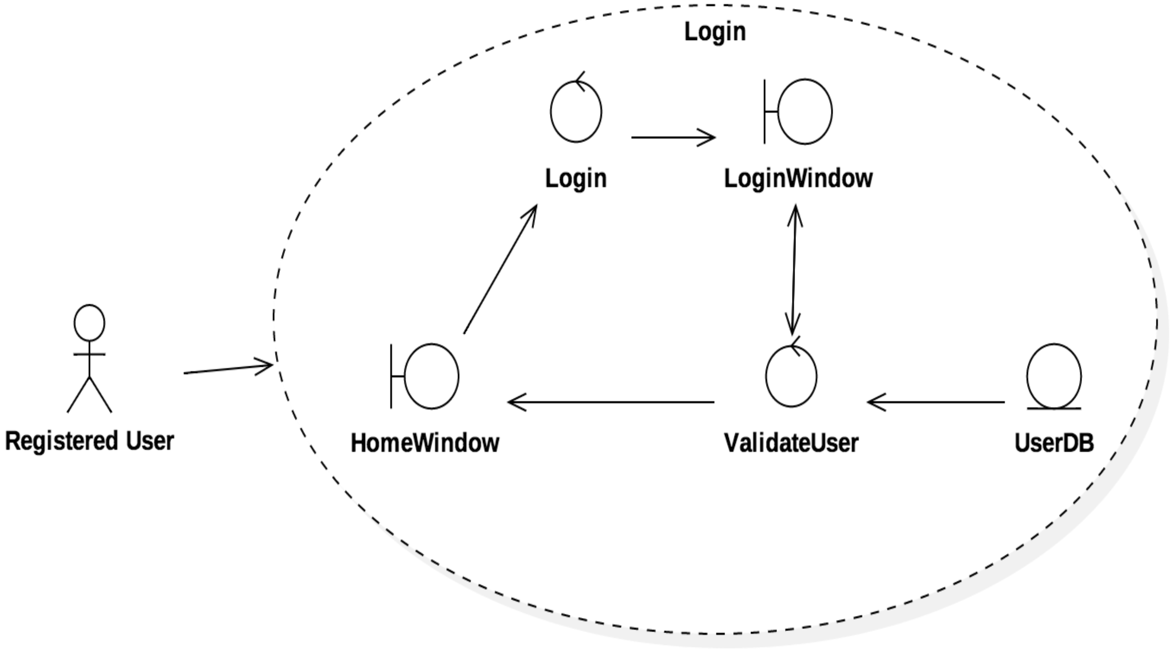


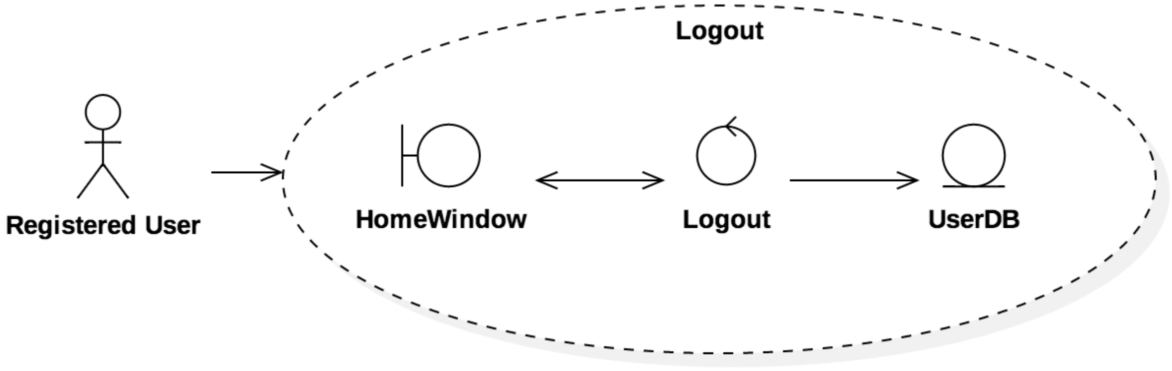

BeaconControl
CloseBeacon
Registered User
BeaconControlWindow
ExtendBeacon
BeaconListings
BeaconCreationWindow
RelocateBeacon


EditProfile
EditProfile
ProfileEditWindow
CancelChanges
Registered User
ProfileWindow
SaveChanges
UserDB

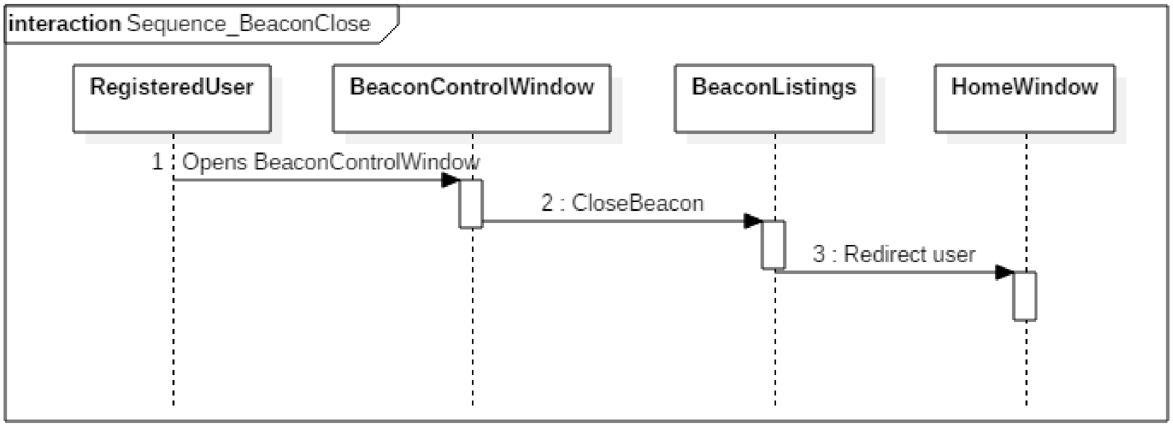

JoinBeacon
ApplySearchFilters
BeaconListings
DisplayResults
SearchWindow
JoinBeacon
BeaconJoinedWindow
Registered User
Guest User
BackToSearch
BeaconFailedJoinWindow

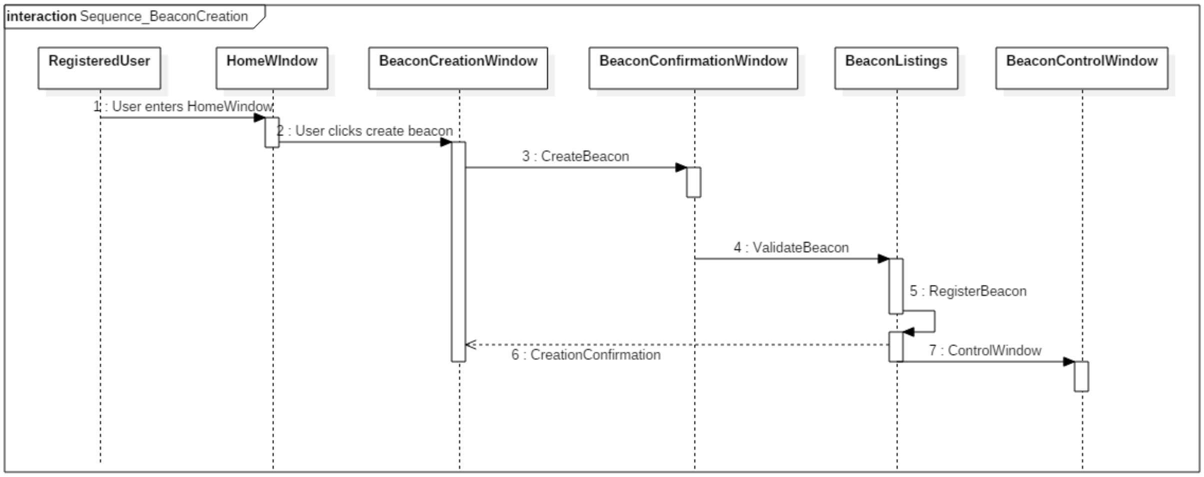

LeaveBeacon
Registered User
BeaconJoinedWindow
LeaveBeacon
HomeWindow
Guest User
BeaconListings

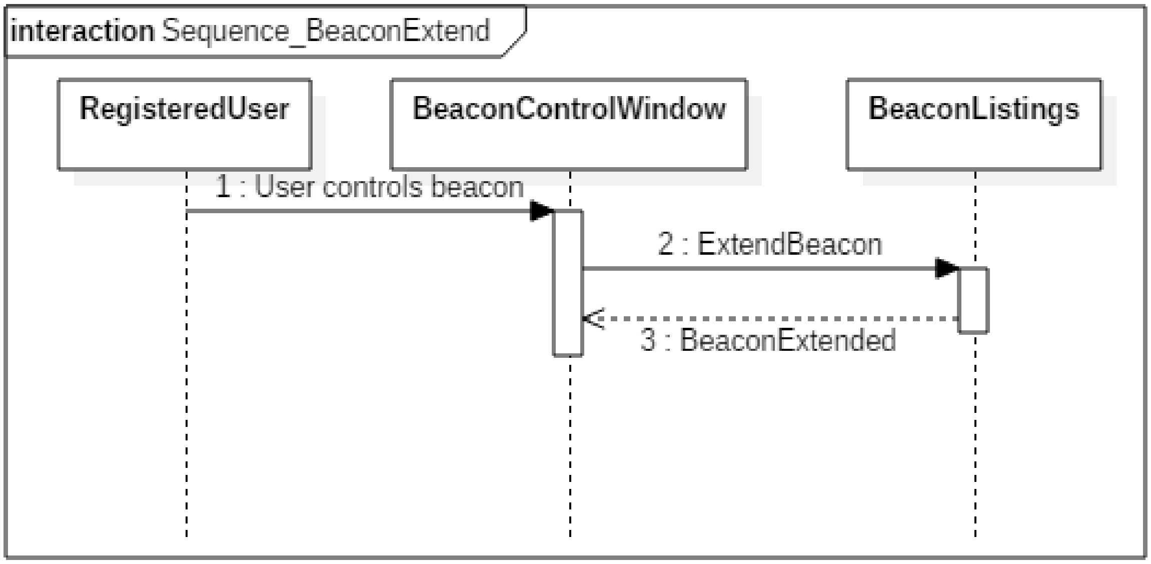


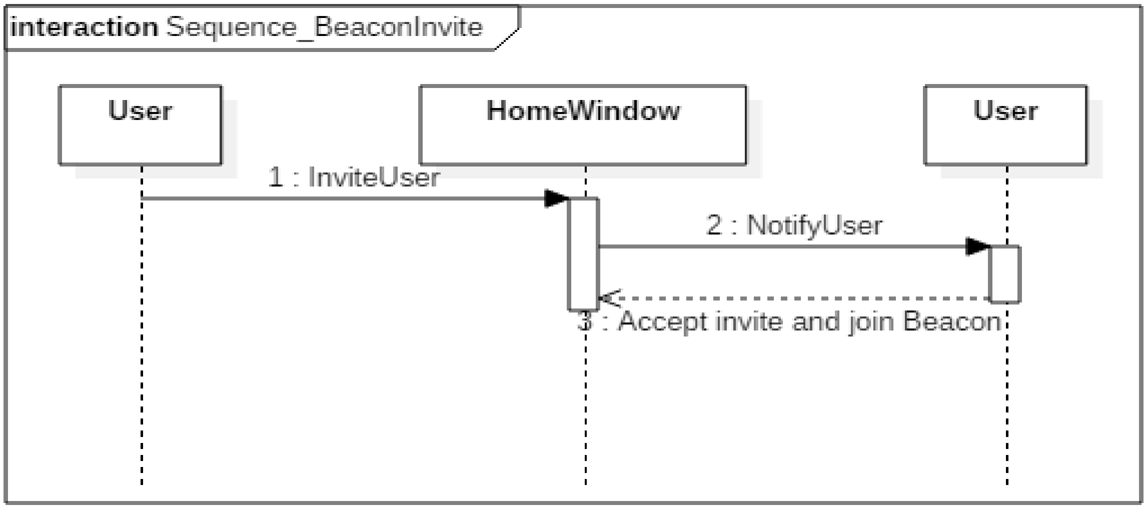


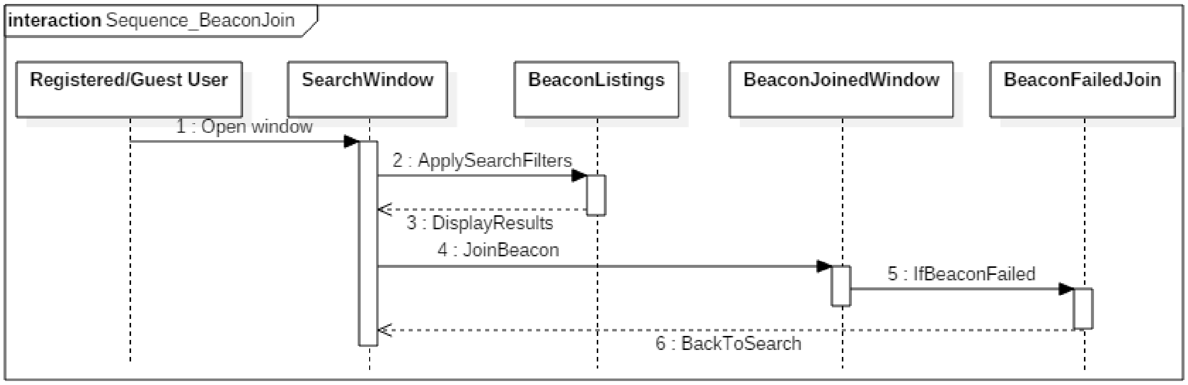
Sequence Diagrams

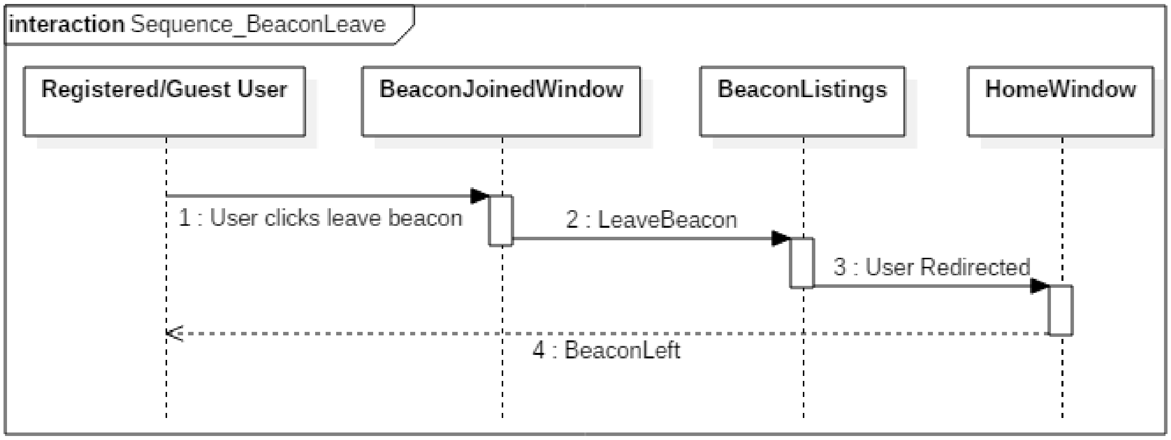


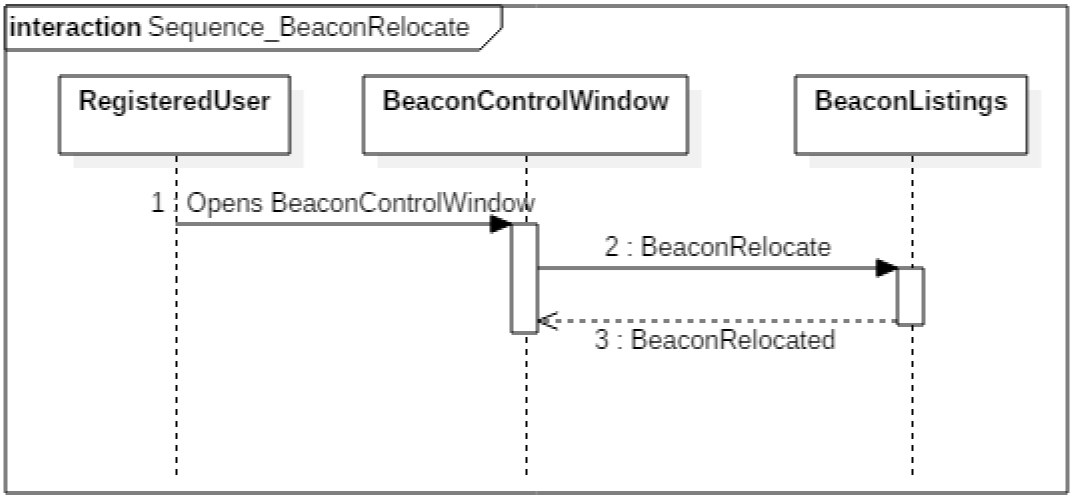

interaction Sequence_BeaconCreation
RegisteredUser
HomeWIndow
BeaconCreationWindow
BeaconConfirmationWindow
BeaconListings
BeaconControlWindow
1 : User enters HomeWindow
2 : User clicks create beacon
3 : CreateBeacon
4 : ValidateBeacon
5 : RegisterBeacon
6 : CreationConfirmation
7 : ControlWindow

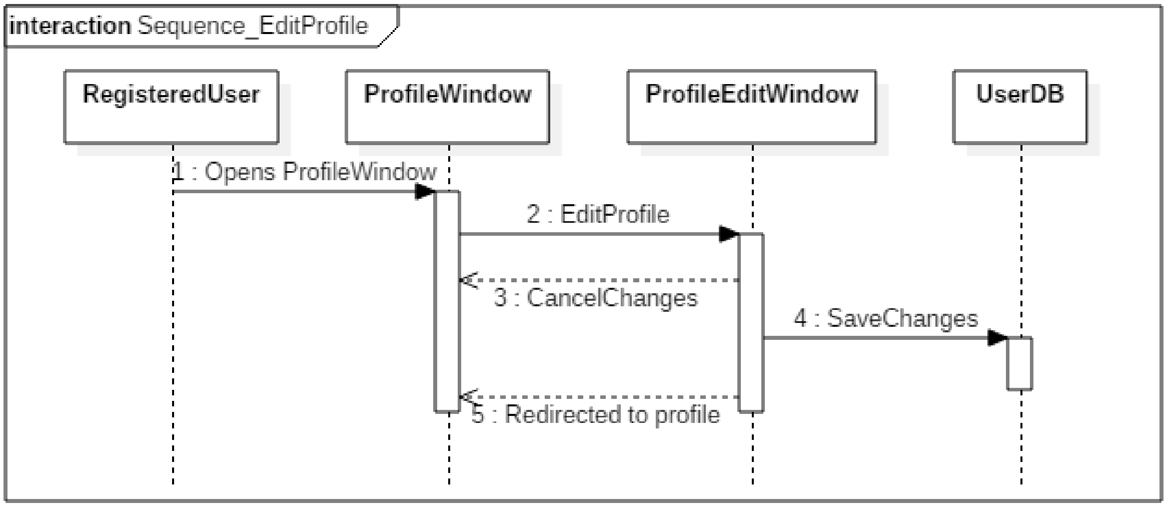

interaction Sequence_BeaconExtend
RegisteredUser
BeaconControlWindow
BeaconListings
1 : User controls beacon
2 : ExtendBeacon
3 : BeaconExtended

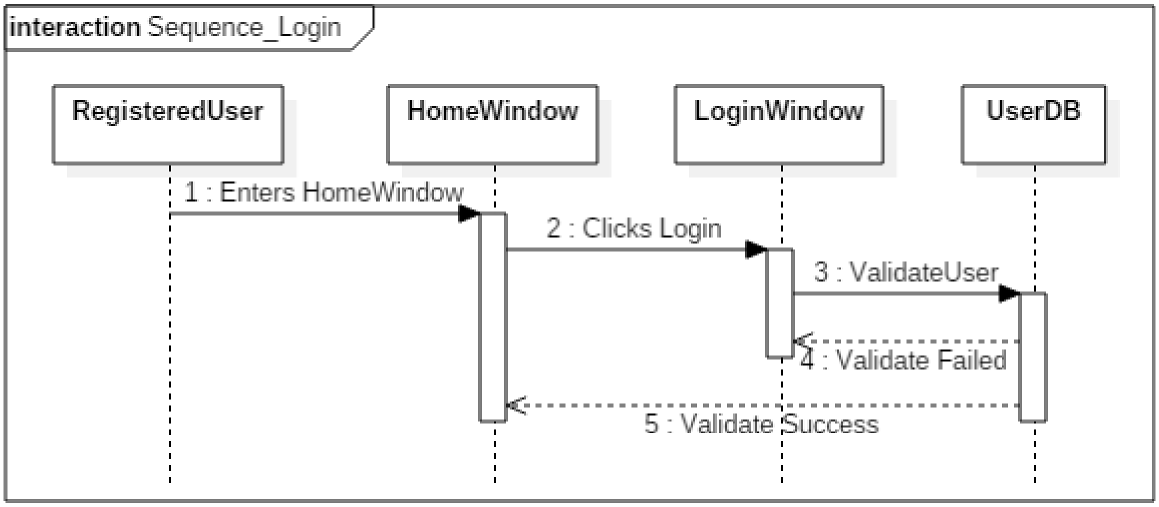

interaction Sequence_BeaconInvite
User
HomeWindow
User
1 : InviteUser
2 : NotifyUser
3 : Accept invite and join Beacon

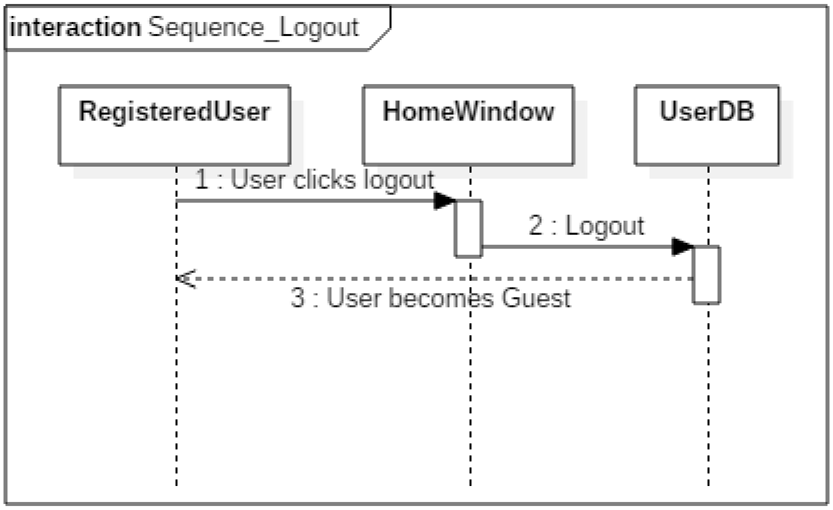

interaction Sequence_BeaconJoin
Registered/Guest User
SearchWindow
BeaconListings
BeaconJoinedWindow
BeaconFailedJoin
1 : Open window
2 : ApplySearchFilters
3 : DisplayResults
4 : JoinBeacon
5 : IfBeaconFailed
6 : BackToSearch


interaction Sequence_BeaconLeave
Registered/Guest User
BeaconJoinedWindow
BeaconListings
HomeWindow
1 : User clicks leave beacon
2 : LeaveBeacon
3 : User Redirected
4 : BeaconLeft


interaction Sequence_BeaconRelocate
RegisteredUser
BeaconControlWindow
BeaconListings
1 : Opens BeaconControlWindow
2 : BeaconRelocate
3 : BeaconRelocated


interaction Sequence_EditProfile
RegisteredUser
ProfileWindow
ProfileEditWindow
UserDB
1 : Opens ProfileWindow
2 : EditProfile
3 : CancelChanges
4 : SaveChanges
5 : Redirected to profile


interaction Sequence_Login
RegisteredUser
HomeWindow
LoginWindow
UserDB
1 : Enters HomeWindow
2 : Clicks Login
3 : ValidateUser
4 : Validate Failed
5 : Validate Success


interaction Sequence_Logout
RegisteredUser
HomeWindow
UserDB
1 : User clicks logout
2 : Logout
3 : User becomes Guest

Authors

Garrett Parris

Ben Bogusat

Gareth Sharpe

Colin Kim