

Guttersnipe (0.3)

*Software
Requirement
Specifications*

Revision History

| Date | Name | Description | Author |
|--------------------|------|-------------|-----------------|
| September 14, 2016 | 0.3 | Draft | Mitchell Verter |

Table of Contents

1. Introduction
 - 1.1. Proposal
 - 1.2. Technical Introduction
 - 1.2.1. Purpose
 - 1.2.2. Scope
 - 1.2.3. Acronyms and Abbreviations
 - 1.2.4. Summary
2. Overview
 - 2.1. Use Case Diagram
 - 2.1.1.
 - 2.2. Collaboration Diagrams
 - 2.2.1. SHAREABLE: READ
 - 2.2.1.1. ViewShareable
 - 2.2.1.2. SearchShareables
 - 2.2.2. ACCOUNT CONTROL: EXTERNAL
 - 2.2.2.1. Login
 - 2.2.2.2. Register
 - 2.2.2.3. RecoverPassword
 - 2.2.3. ACCOUNT CONTROL: INTERNAL
 - 2.2.3.1. EditProfile
 - 2.2.3.2. EditSchedule
 - 2.2.3.3. RenewMembership
 - 2.2.3.4. ChangePassword
 - 2.2.3.5. Logout
 - 2.2.3.6. DeleteAccount
 - 2.2.4. SHAREABLE: CREATE, UPDATE, DELETE
 - 2.2.4.1. AddShareable
 - 2.2.4.2. EditShareable
 - 2.2.4.3. DeleteShareable
 - 2.2.5. SHAREABLE: ANNOTATE
 - 2.2.5.1. RateShareable
 - 2.2.5.2. CommentOnShareable
 - 2.2.5.3. DeleteComment
 - 2.2.6. COMMUNICATIONS
 - 2.2.6.1. SendMessage
 - 2.2.6.2. ReadMessage
 - 2.2.6.3. DeleteMessage
 - 2.2.6.4. BlockUser
3. Class Definitions and Diagrams
 - 3.1. User and Subclasses
 - 3.1.1. User and Subclasses Diagram
 - 3.1.2. User Definition
 - 3.1.3. Guttersnipe Definition
 - 3.1.4. Boss Definition
 - 3.2. Shareable and Parts
 - 3.2.1. Shareable and Parts
 - 3.2.2. Shareable Definition
 - 3.2.3. Time Definition,
 - 3.2.4. Thing Definition
 - 3.3. 3.3.ER Class Diagram
4. Screenshots from 0.2 release and Future Wireframes
 - 4.1. Front Page
 - 4.2. Create Shareable Wizard

- 4.2.1. CreateShareable: Start
- 4.2.2. CreateShareable: Instructions
- 4.2.3. CreateShareable: Describe
- 4.2.4. CreateShareable: Classify (I)
- 4.2.5. CreateShareable: Classify(2)
- 4.2.6. CreateShareable: Map
- 4.2.7. CreateShareable: Schedule (1)
- 4.2.8. CreateShareable: Schedule (2)
- 4.2.9. CreateShareable: Schedule (3)
- 4.3. SearchShareable
 - 4.3.1. SearchShareable: ResultList
 - 4.3.2. SearchShareable: ResultCalendar
 - 4.3.3. SearchShareable: ResultMap
 - 4.3.4. SearchShareable: SearchByCategory
 - 4.3.5. SearchShareable: SearchByTag
 - 4.3.6. SearchSharable: SearchByLocation
 - 4.3.7. SearchShareable: SearchByTime
- 4.4. Authentication
 - 4.4.1. SignIn
 - 4.4.2. SignUp
- 4.5. Documentation
 - 4.5.1. Mission Page
 - 4.5.2. FAQ
 - 4.5.3. Presentation (2013)
 - 4.5.3.1. Start
 - 4.5.3.2. Objective
 - 4.5.3.3. Audience
 - 4.5.3.4. Consultations
 - 4.5.3.5. Other Sites
 - 4.5.3.6. Q: Method
 - 4.5.3.7. A: Method
 - 4.5.3.8. Example
 - 4.5.3.9. REprot
 - 4.5.3.10. Result
 - 4.5.3.11. Research
 - 4.5.3.12. Fin
 - 4.5.4. Administrative
 - 4.5.4.1. Legal
 - 4.5.4.2. About
 - 4.5.4.3. Contact
 - 4.5.5. Kropotkins
 - 4.5.5.1. Kropotkin Quotes
 - 4.5.6. Screensize
 - 4.5.6.1. Dropdown top menu (1)
 - 4.5.6.2. Droptown to menu (2)

Part 1: PROPOSAL

1. What is the site/app?

Guttersnipe is a web portal and mobile app that caters to anarcho-communist street youth (and adults) who desire to subvert capitalism by sharing resources.

It will enable people to broadcast to each other locations of shareable resources, distributed among four main categories:

1. Housing: squats, abandoned buildings, punk houses, etc.
2. Food: dumpsters, Food Not Bombs, free meals, etc.
3. Healthcare: clinics, needles, condoms, etc.
4. Movement: rideshares, train maps

Eventually, other types of resource sharing will be integrated into the application.

Each Shareable will be characterized as a

1. Thing: Categorization and Tags
2. Space: Geolocation
3. Time: Schedule

Further development will require a close study of the writings of Kropotkin and Fourier.

2. What need does this meet? or problem does it solve?

This application serves the urgent need to overthrow capitalism by helping people to self-organize outside and beyond the market of commerce.

The ultimate intention is to facilitate the creation of alternate avenues of exchange, freely organized by free individuals.

3. Who is going to go/use to this site/app?

In the current incarnation, it is mostly aimed towards the freegans gutterpunks, who live off of dumpstered food, live in squatted housing, and travel by hopping trains.

As we get a better sense on the needs of the anticapitalist community and possibilities for alternative organizing, we will expand the possibilities for anti-market resource sharing.

4. Why will they go to this site/app?

To find food, clothing, shelter, etc.

5. Why will they keep coming back to your site/app?

See above.

6. How is it different from other similar sites?

There are similar sites of various types, but many of them have certain faults.

There is a site called rideshare.com; there is a site named couchsurfer.com; there is freecycle.com, which allows the sharing of goods.

These are all laudable efforts. Some of these are marred by an underlying desire for profit. But some of them are motivated out of genuine desire to promote Mutual Aid.

The very mission of Guttersnipe.net will be to promote the organization of the lumpenproletariat and to create alternative exchanges outside of capitalism. This mission will enable Guttersnipe.net to be singularly focused on this goal.

It will thus be able to bring together whatever resources necessary for the undermining of capitalism: the various services— such as squatting, dumpster diving, hitchhiking, train hopping, resource sharing, etc — will be coordinated on a singular web portal.

In addition, there are several web portals that are dedicated towards the promotion of anarcho-communist goals..

Such sites are

- Freegan.info
- Picture the Homeless
- Squat.net
- Foodsharing (Germany)

Many of our initial design specifications will be taken from the freegan group and Picture the Homeless.

In addition, we intend Guttersnipe to be cross platform, available both via the web and as a mobile app.

To my knowledge, there are not yet any apps dedicated with such a task.

7. What steps will a person go through interacting with the site/app?

Most of the various interactions will be handled using forms.

The various services offered by Guttersnipe all boil essentially boil down to two types of transaction:

1. information submission;
2. information retrieval.

One person posts about an abandoned building or a good dumpster; another person searches for such information.

** All Users can view a Shareable or search all Shareables.

Shareables can be searched and results will be shown with the following data:

1. Thing: Description, Categorization, Tags
2. Space: Map
3. Time: Calendar

** Registered Users (known as “Guttersnipes”) can add, edit, and delete Shareables. Guttersnipes may also rate Shareables, comment on Shareables, and erase these comments.

Guttersnipes can manage their profiles, which contains their availability schedules, names, account expiration date, optional email, optional password, optional location, and optional contact info.

All Guttersnipe user accounts expire after a certain date, but this date may be extended at any time.

Guttersnipes can communicate to each other messages that contain Schedules and Text in order to coordinate a meeting time.

**** Bosses**

Bosses are Guttersnipes with administrative capacities.

They can delete any Guttersnipe account, any Shareable, and any Comment.

Practical Constraints

Security

We will have to build in a security infrastructure in the project in order to guarantee anonymity of transactions.

Tor will be used to anonymize transactions.

Whisper will be used to encrypt communications and interactions.

The host server will have to be able to run Python/Flask.

The client will have to have an accessible webview for the deployment of Javascript.

We will have to design a User Interface that supports a 1.8 inch QQVGA (128x160) Display to support the phones provided by government assistance <http://newsroom.assurancewireless.com/custom-page/product-information>

1.1. Purpose

Guttersnipe promises to be a platform for individuals and groups to freely share resources such as food, shelter, and medicine.

This Document will detail the features of Guttersnipe, and will serve as a guide to developers, and as a legal document and users manual for prospective clients.

1.2. Scope.

1.2.1. Users

1.2.1.1. User

All System users can search the Shareables view a single Shareable, and read the rating and the comments ascribed to it.

1.2.1.2. Vagrant: This class represents all visitors to the site who has not yet signed in as a member.

The Vagrant class inherits all the properties and functionalities of the base class User.

All Vagrants may register to become a user, may login as a user, and may retrieve a lost username or password.

1.2.1.3. Guttersnipe: This class represents a user who has registered for an account in the system.

This class inherits all the properties and functionalities of the base class User.

In addition, the Guttersnipe can exercise control over its own account. The Guttersnipe may edit its own profile, its own location and edit its own availability schedule.

It may renew its membership and change its password.

It may logout of its account.

The Guttersnipe may also create a new Shareable, and edit or delete a Shareable that it has created.

The Guttersnipe may rate or comment on a Shareable and may delete a previous comment.

The Guttersnipe may send messages and send the schedules of other

1.2.1.4. Boss: The Boss class represents the administrative users of the System.

Bosses have all the same properties and functionality as the Guttersnipe, but their functionalities are unlimited in scope.

The Boss may edit or delete any Shareable, may delete any User, and may delete any comment.

1.2.2. Business Objects

1.2.2.1. Shareable: Each shareable is classified according to its categorization and description, its schedule, and its location.

1.3. Acronyms and Abbreviations

1.3.1. V: Vagrant

1.3.2. G: Guttersnipe [User]

1.3.3. B: Boss

1.3.4. SRS: Software Requirements Specification

1.3.5. GUI: Graphical User Interface.

1.3.6. FSM: Finite State Machine.

1.3.7. 1 DB: Database.

1.3.8. 1 ERCD: Entity-Relation Class Diagram.

1.4. Summary

The rest of this SRS is organized as follows:

Section 2: Gives the overall description of the Guttersnipe application. It contains the Use-Case diagram and descriptions for Guttersnipe. Section 2 also contains the assumptions and dependencies of the system.

Section 3: Gives specific software requirements and functionalities in the form of Mini Use- Case diagrams along with accompanying Collaboration diagrams, Finite State Machine of the system, and ER Class diagram of the system. This section also contains supplementary software requirements of the systems.

Section 4: GUI Components: The appendix contains user interface prototypes for the system, including many screenshots from the 0.2 release of the application.

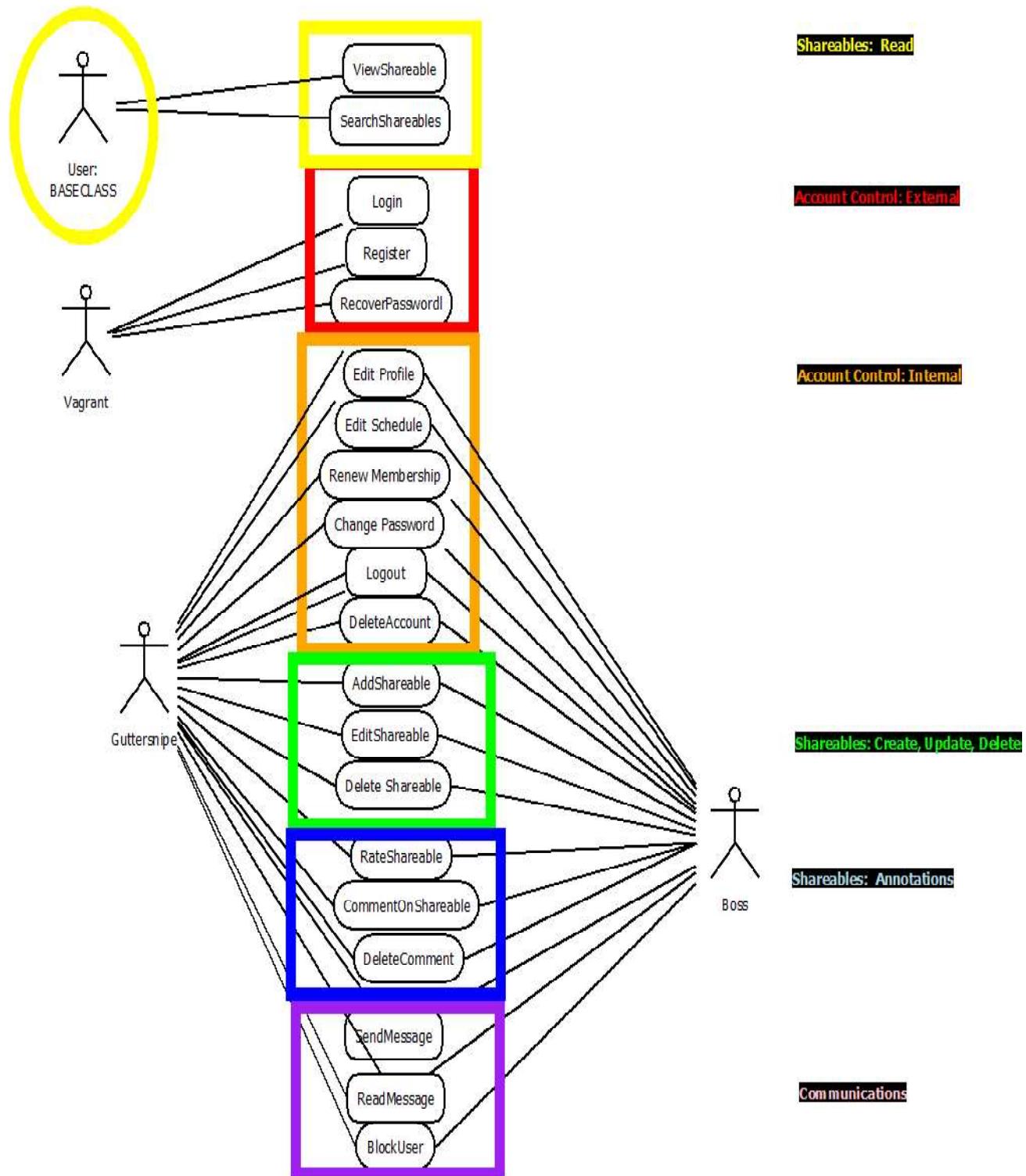
02: Overview

02.01: Use Case Diagrams and Descriptions

02.02: Server/Client Assumptions and Dependencies

2. Overview

2.1 User Case Diagram and Descriptions



02.01.01

Usecases: USERS

User (U)

Shareables (Read)

View: Any U may view a search result.

SearchShareables: Any U may search for Shareable by Thing (Categorization, Tags), Space (Geolocation), and Time (Schedule)

Vagrant (V)

Account Management (External)

Login: Any V can login to the system, which transforms V into a G or B .

Register: Any V may register to become a G.

RetrieveUsername: Any V may request a username reminder.

RecoverPassword: : Any V may request a password reminder.

Guttersnipe (G)

Account Management (Internal)

EditProfile: Can edit information on own account.

EditSchedule: Can edit availability schedule.

RenewMembership: Can renew the terminal date of membership

Logout: Can log out of own account.

ChangePassword: Can reset own password.

DeleteAccount: Can delete own account.

Shareables (Create, Update, Delete)

CreateShareable: Can add a Shareable resource.

UpdateShareable: Can update Shareable created by self.

DeleteShareable: Can delete Shareable created by self.

Shareables (Annotate)

RateShareable: Can rate any shareable

CommentOnShareable: Can comment on any shareable.

DeleteComment Can delete own comment.

Communications

SendMeetup: Can communicate message (schedule + text) from other

Guttersnipe.

ReadMessage: Can read message inbox.

SetSchedule: Can set availability calendar.

BlockUser: Can block any other user except Boss.

Boss (B)

Boss has same capabilities as Guttersnipe, but they are unrestricted to apply to all system users. The relevant overrides are as follows:

UpdateShareable: Can update Shareable created by any user.

DeleteShareable: Can delete Shareable created by any user.

Deleting a Shareable account triggers DeleteAccount, DeleteComments and DeleteShareables for that User.

02.02.01

Client / Server Assumptions and Dependencies

Previous release

The previous release (0.2.1.5) of Guttersnipe was a MEAN stack application, utilizing Mongo/Mongoose, Node, Express, and Angular 1.x.

The deployment of a node application requires a server or PAAS that supports a node engine. We find this constraint too restrictive because many servers do not have a node engine, including the servers where we hope to do our initial deployments.

Although Mongo/NoSql is a fine technology, we believe that rapid, optimized database queries can best be done with an SQL database, which will obviate the need for a lot of the “middle-tier” post-processing of results from a database query.

Current release

Client. The client requires a contemporary browser that supports standard HTML 5, CSS 3, and Javascript. It will use Angular 2.x as a front-end framework.

The application will be ported to mobile devices initially by taking advantage of libraries like phonegap which allow one to use the device webview to deploy web interfaces. Native Android/ iOS ports may be attempted as well.

For backend technologies, we sought a technology that could easily be ported to a variety of servers where no superuser access is required. We dislike bulky frameworks, preferring to add components as we need them. To this end, we have chosen the Flask / Python framework.

The most important platform to develop upon are the free phones given by government assistance (<http://newsroom.assurancewireless.com/custom-page/product-information>). Some of these have 1.8 inch screens. We are currently unsure of what toolkit we will need to use for the front end.

SPECIFIC REQUIREMENTS

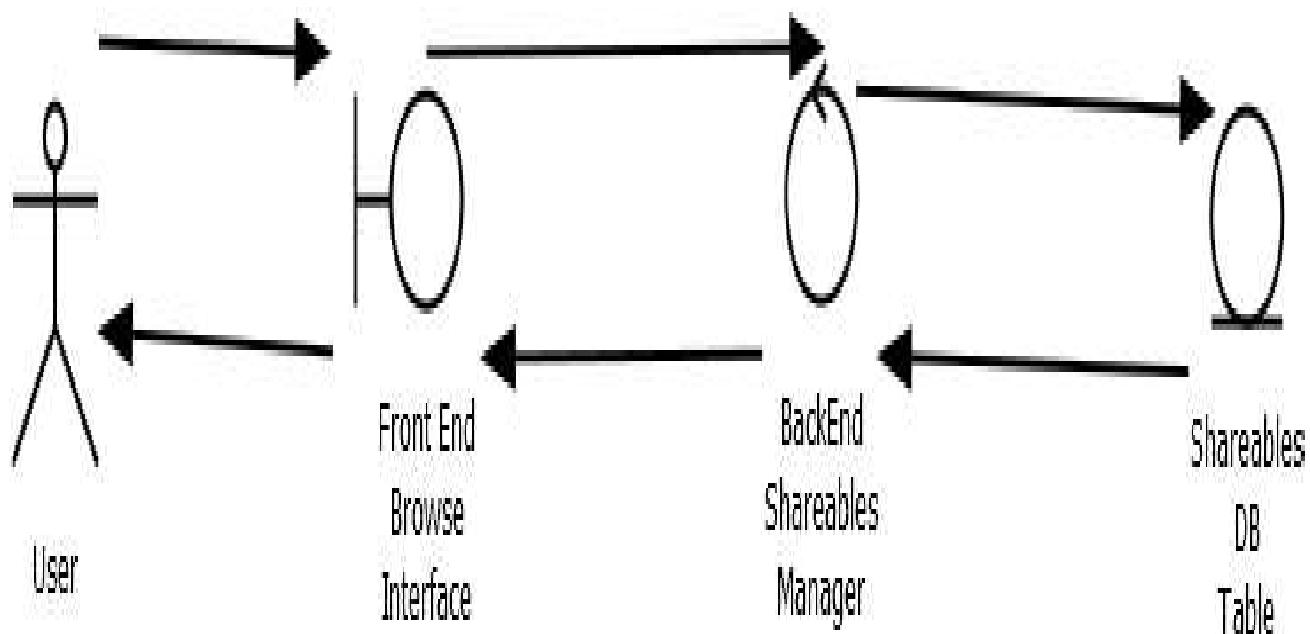
Collaboration Diagrams

3.2.1. SHAREABLE: READ

03.01.04.xx

View Shareable

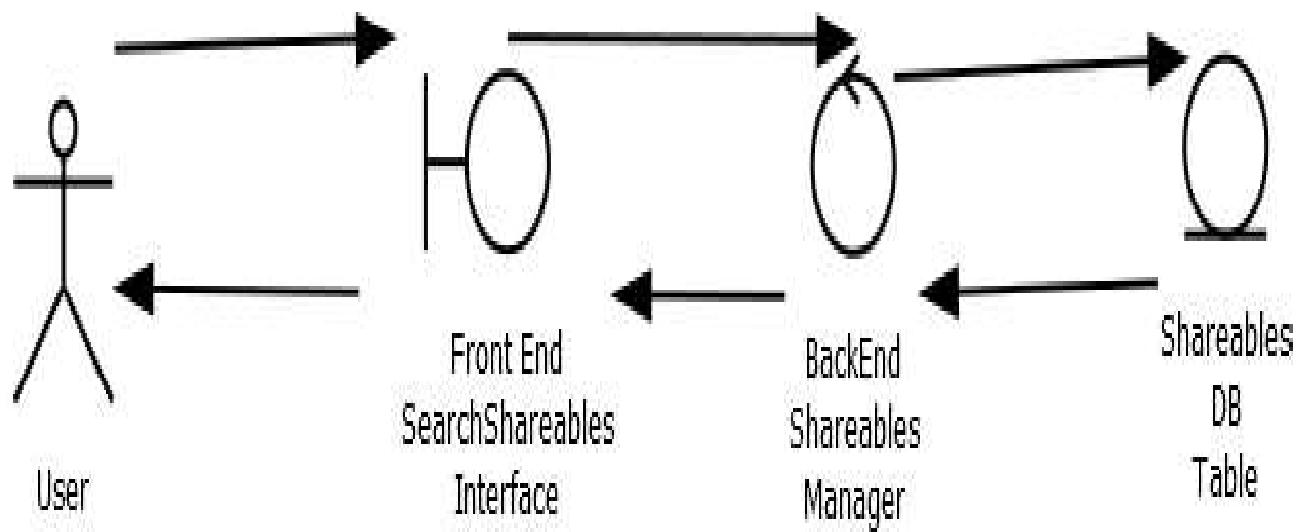
Collaboration Diagram



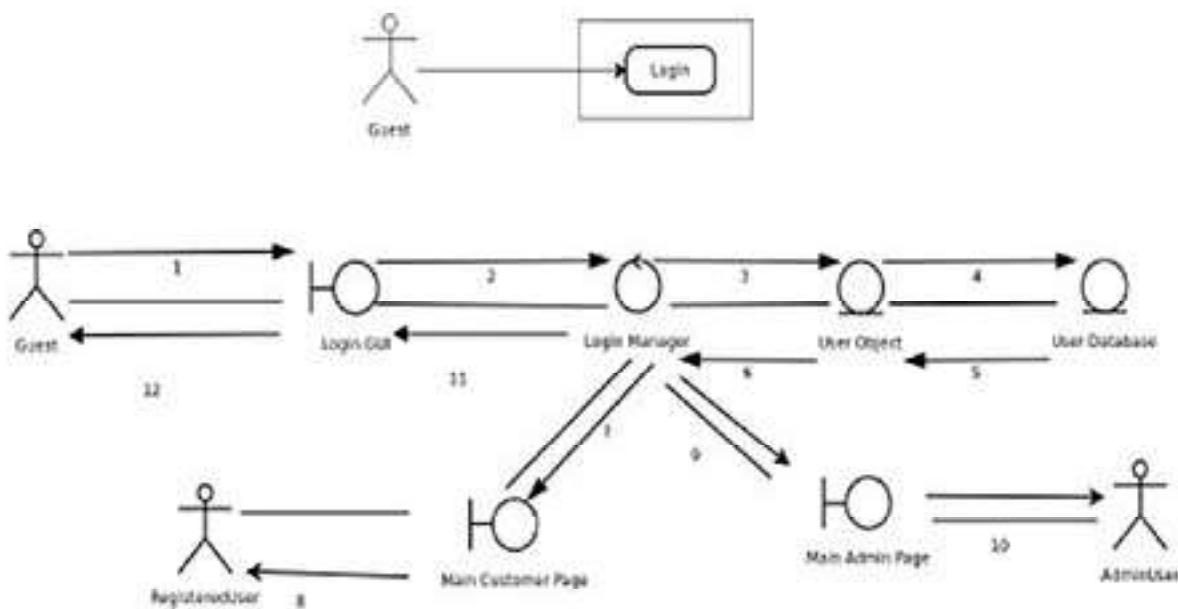
03.01.04.XX

Search Shareables

Collaboration Diagram



3.2.2. ACCOUNT CONTROL: EXTERNAL



1. Guest User fills in account information on the login page
2. User info passed to login manager
3. Pass login info to user object
4. Login object queries user database with login info
5. User database passes back query results to user object
6. Login manager receives results
7. Login manager shows main customer page
8. System recognizes end user as Customer
9. Login manager shows main admin page
10. System recognizes end user as Admin

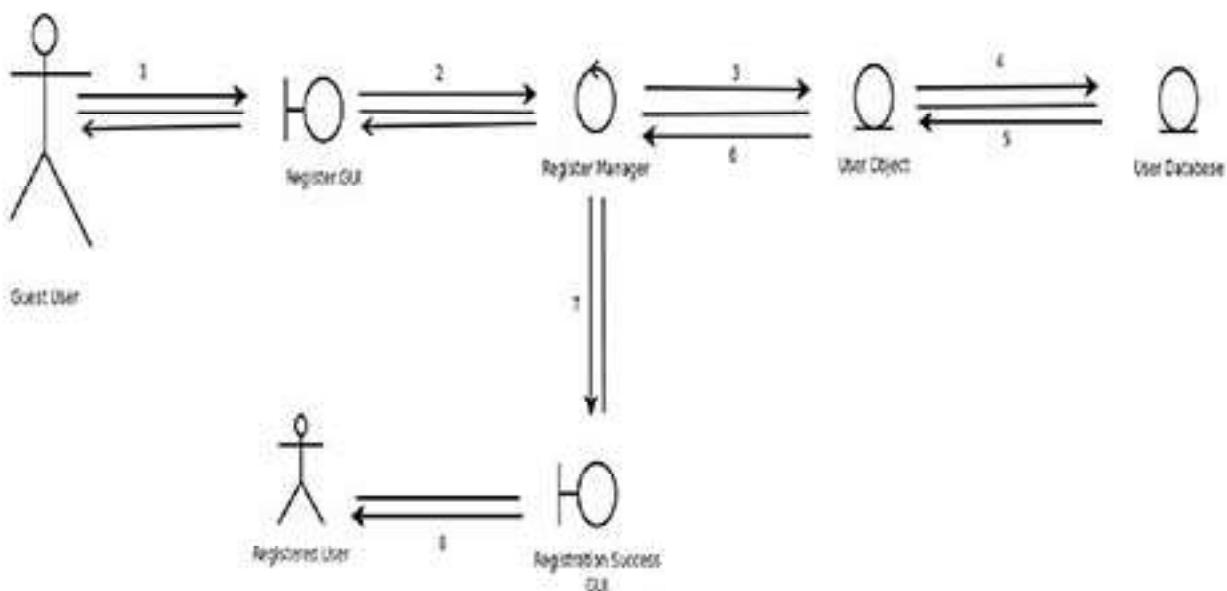
Exception 1

5. DB error sent to Login Object
6. Failure sent to Login Manager
11. Login manager presents Failure message on GUI
12. Guest receives error message

Exception 2

7. First time login user present with choose interest page

Figure B2: Login



1. Guest User clicks on 'Register GU' to be registered as a user.
2. The information is then passed to the register object.
3. This information is plotted to the user object and generates password.
4. The information is stored in the user DATABASE.
5. User database send the information to user object.
6. The correct information passed from the user object is then passed to register object.
7. This opens a new GUI and welcomes the new user.
8. This turns a Guest User be a Registered User.

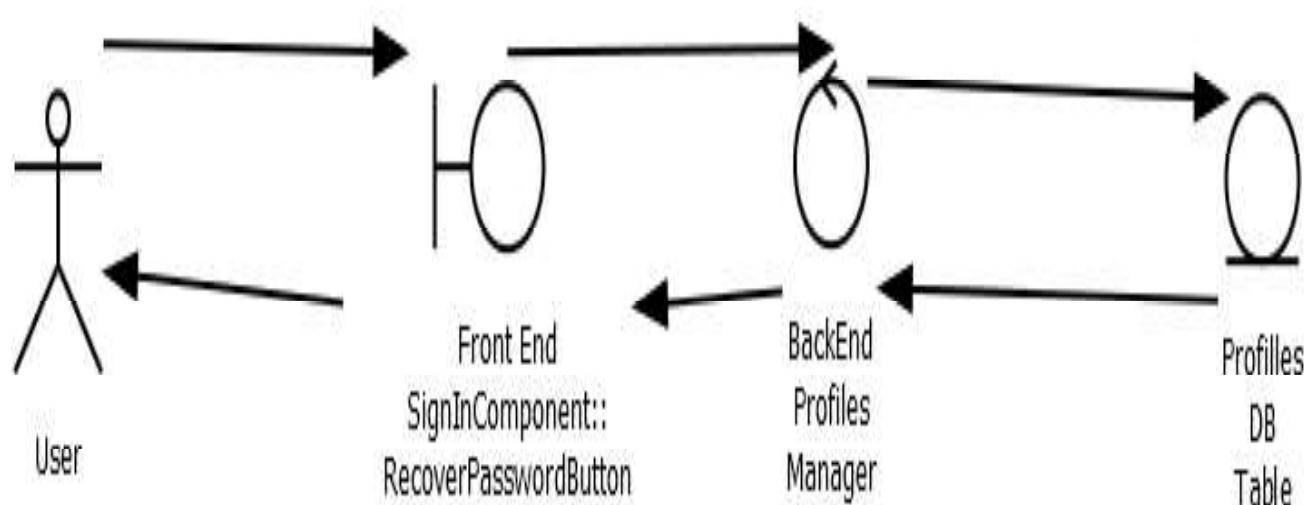
Exceptional Case:

1. The visitor enter the information and click 'Submit'.
2. The information is passed to Register Object.
3. The processed information is wrong so is returned to Register GUI.
4. Guest User is specified the information provided is wrong.

B1: Register

03.01.04.XX

Recover Password Collaboration Diagram

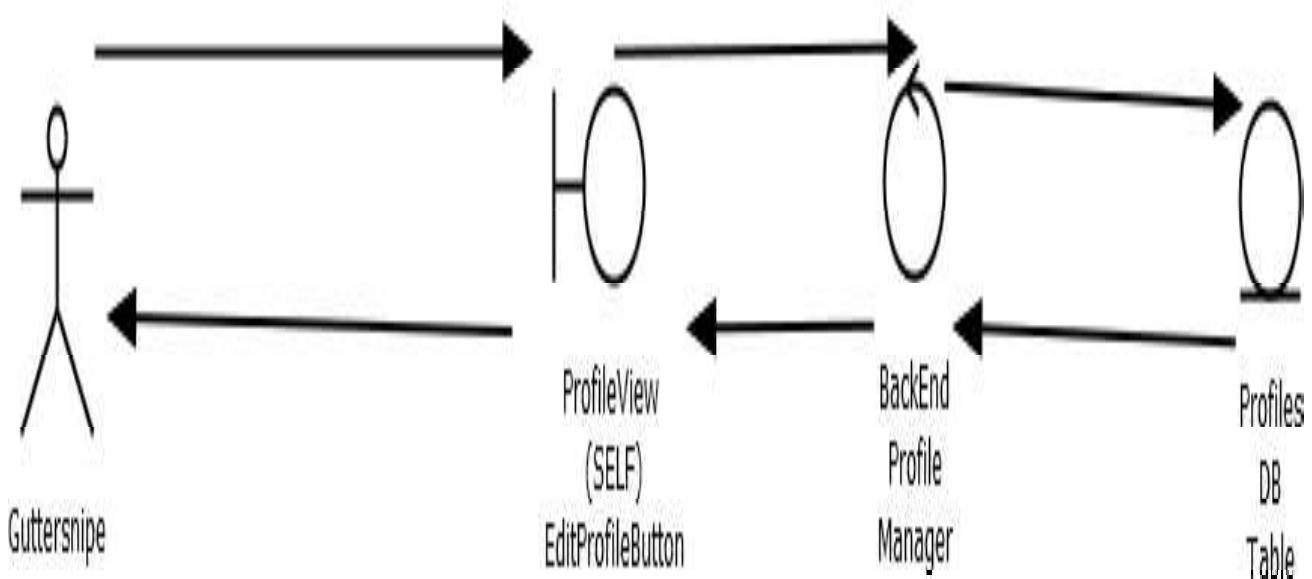


3.2.3. ACCOUNT CONTROL: INTERNAL

03.01.04.XX

Edit Profile

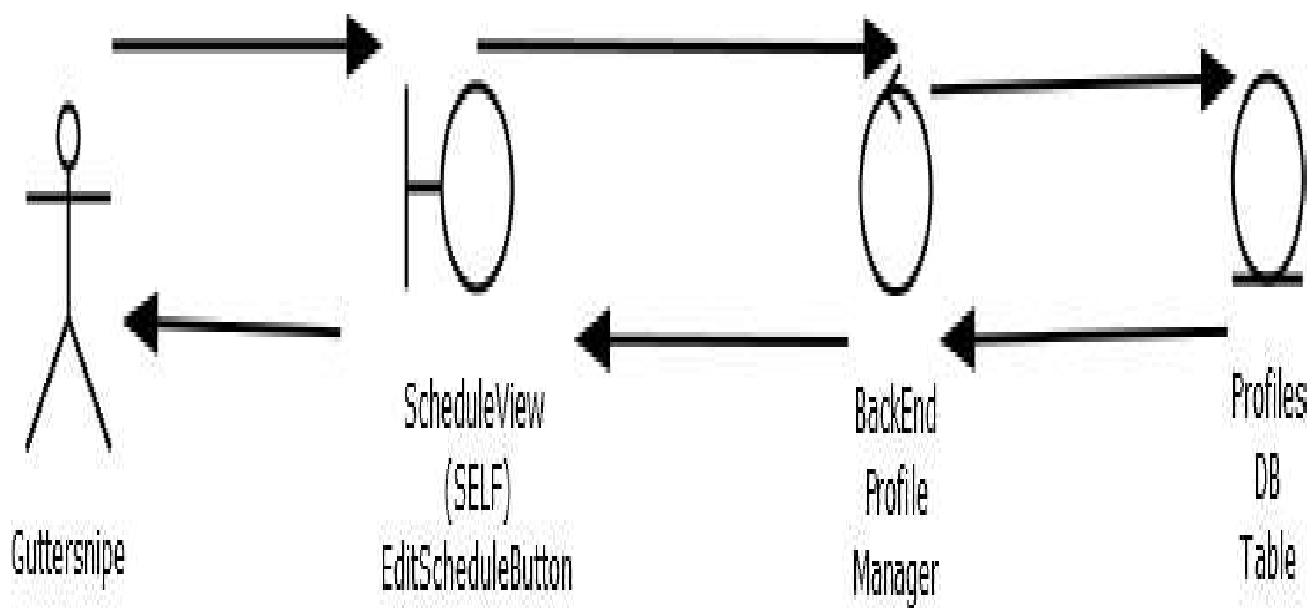
Collaboration Diagram



03.01.04.XX

EditSchedule

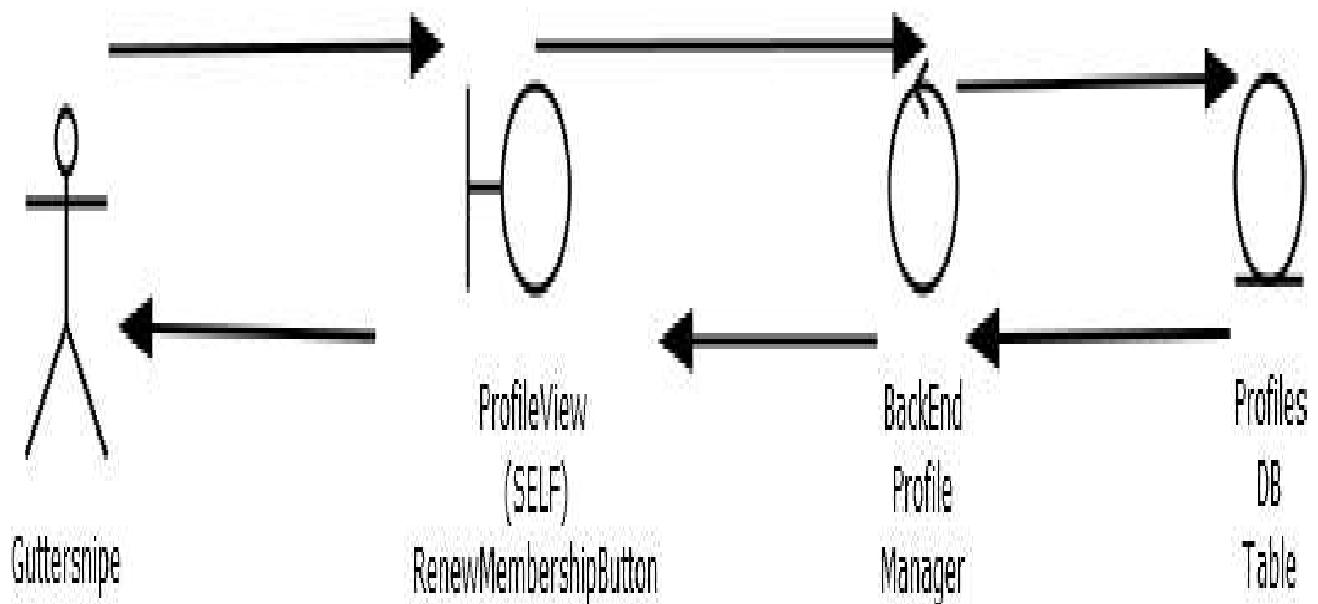
Collaboration Diagram



03.01.04.01

Delete Shareable

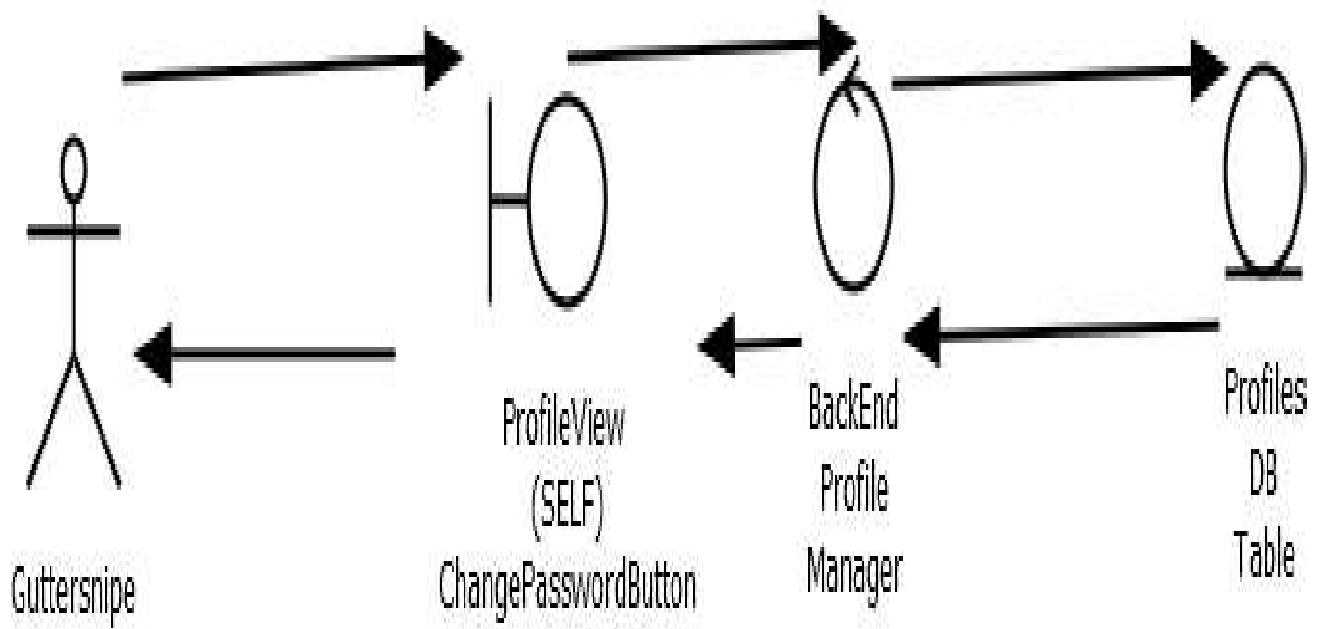
Collaboration Diagram

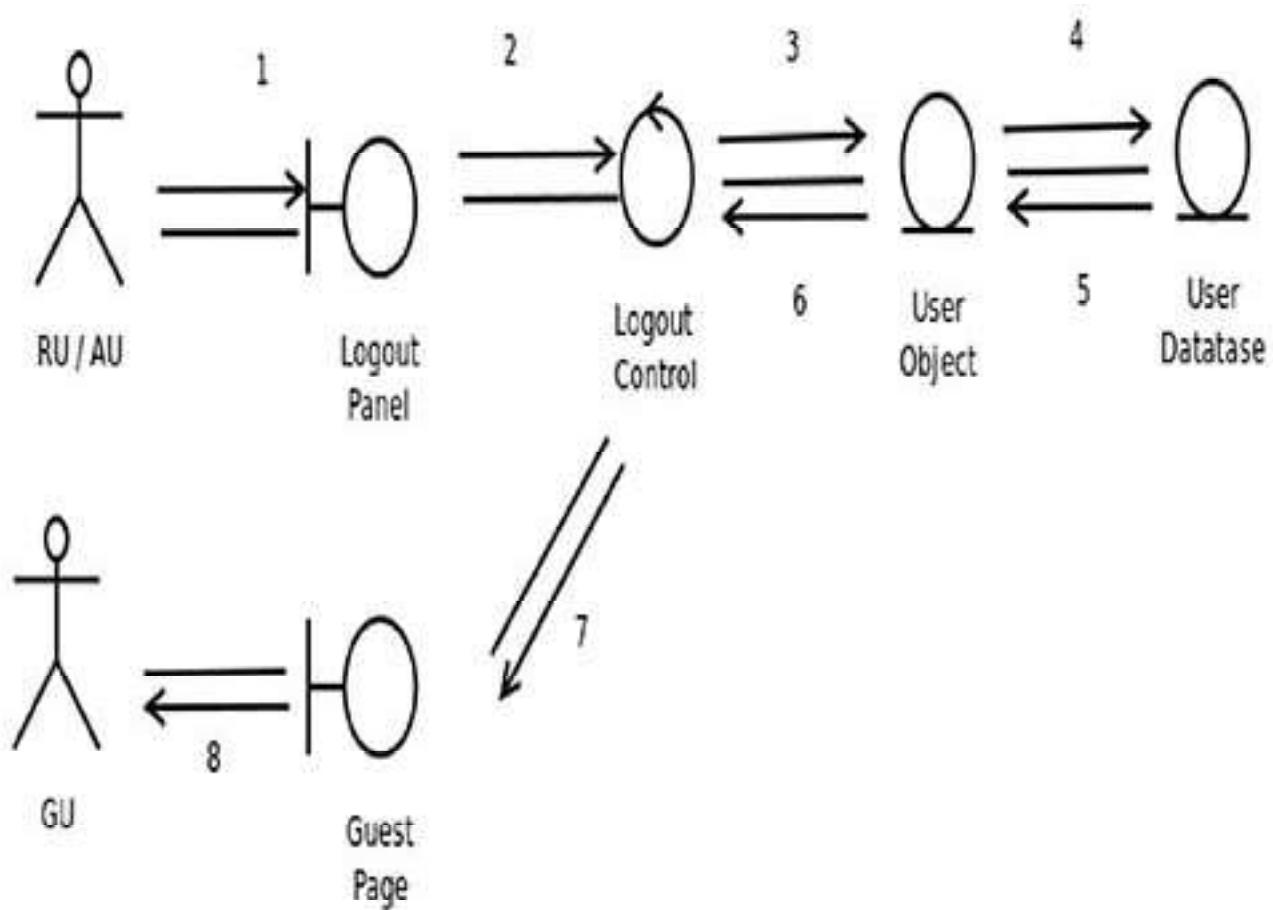


03.01.04.XX

Change Password

Collaboration Diagram





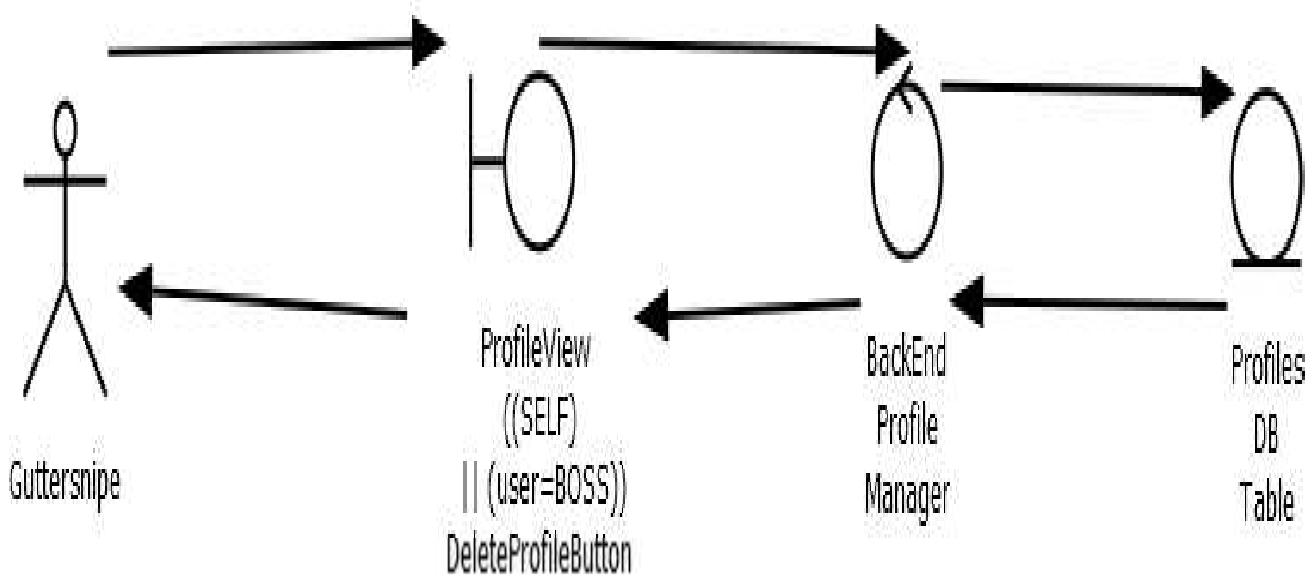
1. Registered User/Admin User clicks on the Logout button
2. Logout Control processes request, verifies user's desire to logout
3. Controller passes logout to user database, ends user's session, completes any pending transactions
4. Database returns results to Logout Control
5. Logout Control produces Guest page stating "You have now been logged out"
6. End User is now considered to be a Guest

Figure C8: Logout

03.01.04.XX

Delete Profile

Collaboration Diagram

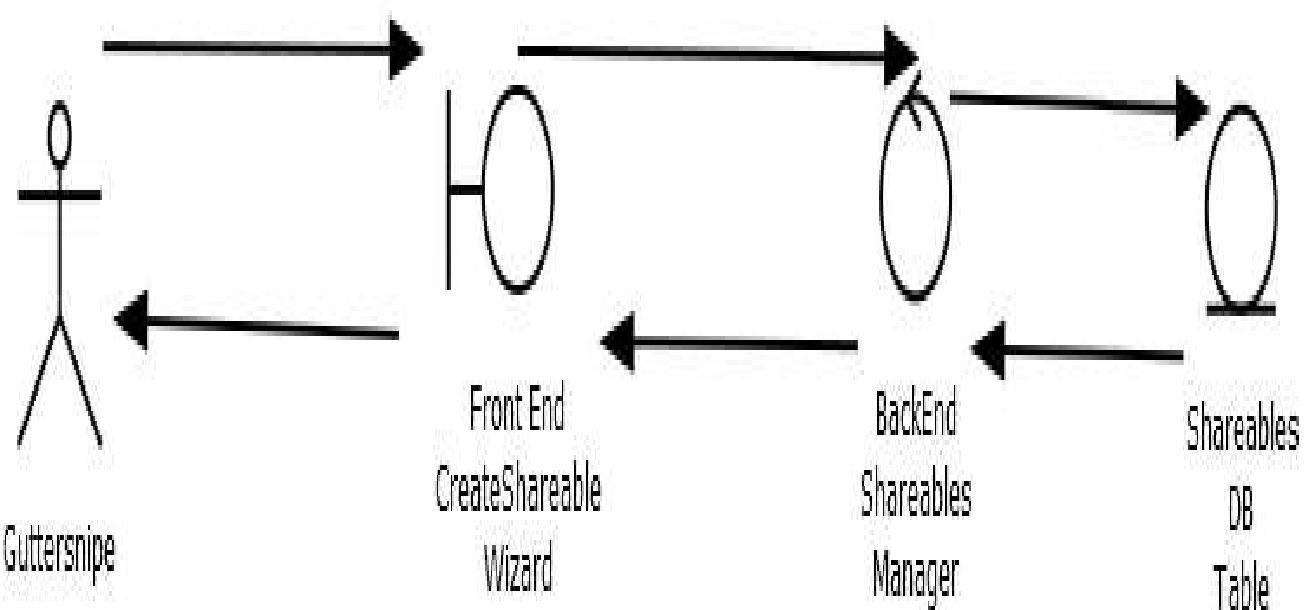


3.2.4. SHAREABLE: CREATE, UPDATE, DELETE

03.01.04.01

Delete Shareable

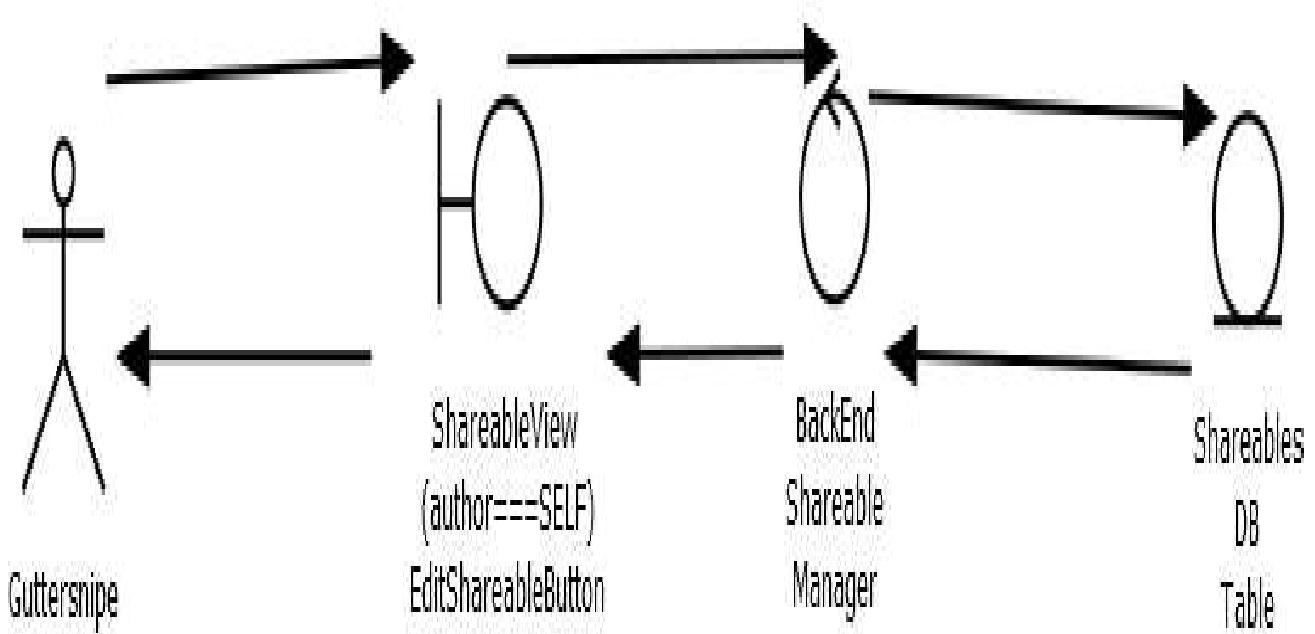
Collaboration Diagram



03.01.04.XX

Edit Shareable

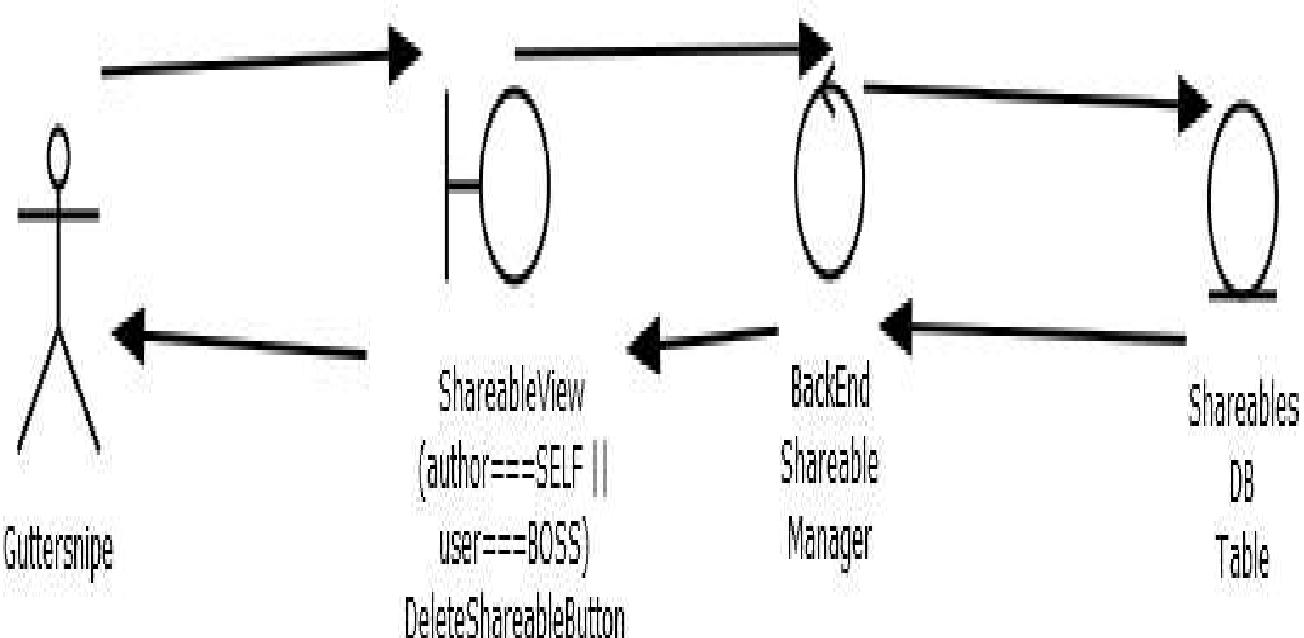
Collaboration Diagram



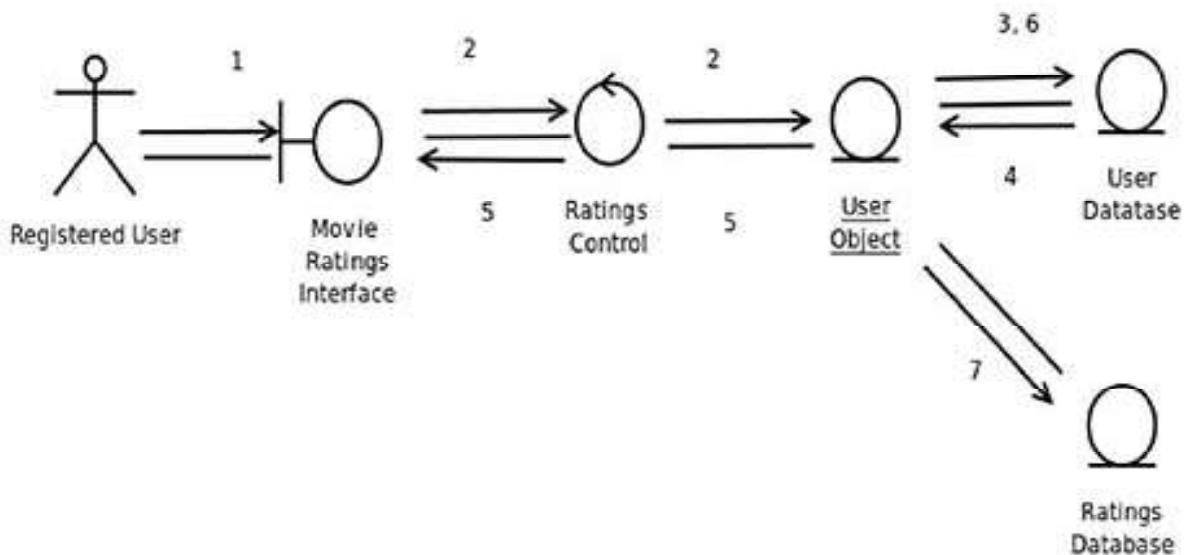
03.01.04.XX

Delete Shareable

Collaboration Diagram



3.2.5. SHAREABLE: ANNOTATE



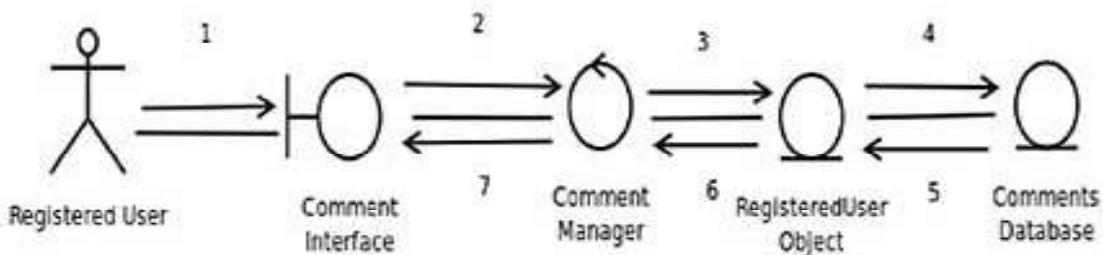
Figure

1. Registered User clicks on movie rating interface
2. Rating Interface submits UserID, MovieID, and Rating to Rating Control
3. Rating Control checks User DB to check User's ratings permissions
4. User DB returns RU's permissions
5. Ratings Control sends success or failure message back to Ratings GUI
6. Ratings Control updates Registered User Database to record Rating Behavior
7. Ratings Control updates Movie Database with new rating

Exceptions:

1. RU presses submit button without selecting a rating
2. RU does not have permissions to rate movie

Figure C2: Rate



1. RU visits the comment interface of a movie page
2. RU submits comment to be processed by the comment manager
3. The Comment Manager accesses RegisterUser object to processes the comment with a timestamp
4. The data is inserted to Comments Database
5. The comment information is sent back to the RegisteredUser object
6. The comment information is passed back to the comment manager
7. The Comment Manager updates the Comment Interface with new comment(s)

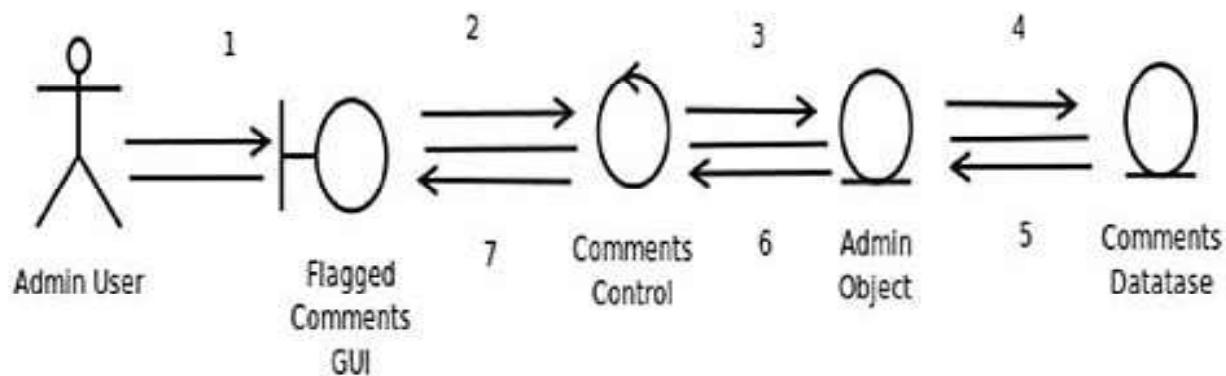
Exception 1:

2. Blank comment is submitted
7. The comment manager displays an error message to the GUI

Exception 2:

2. A comment that exceeds the character limit is submitted
7. The comment manager displays an error message to the GUI

Figure C3: Comment



1. Admin User selects comments for deletion from Flagged Comments Panel.
2. GUI sends deletion request to Comments Control.
3. Comments to be deleted are sent to Database.
4. Comments are sent to comments database.
5. Database returns remaining Flagged Comments from Database.
6. The remaining flagged comments are sent to comments control.
7. The remaining Flagged Comments are displayed on the GUI.

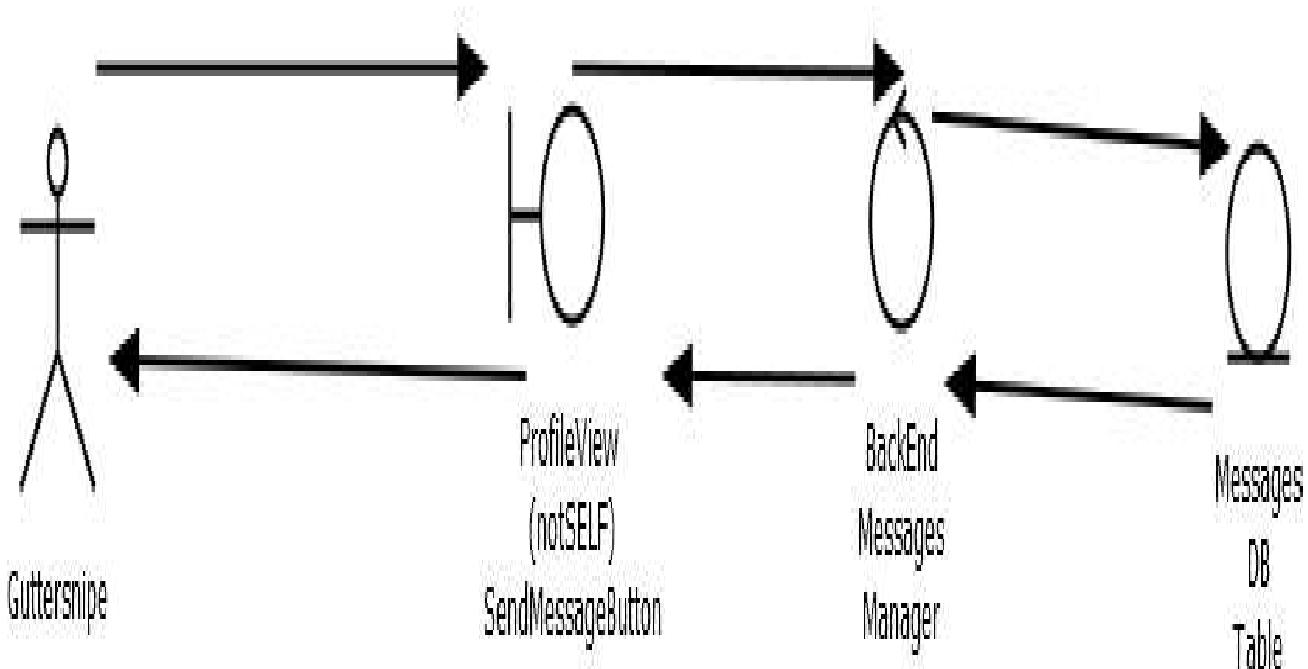
D1: Erase Comment

3.2.6. COMMUNICATIONS

03.01.06.xx

Send Message

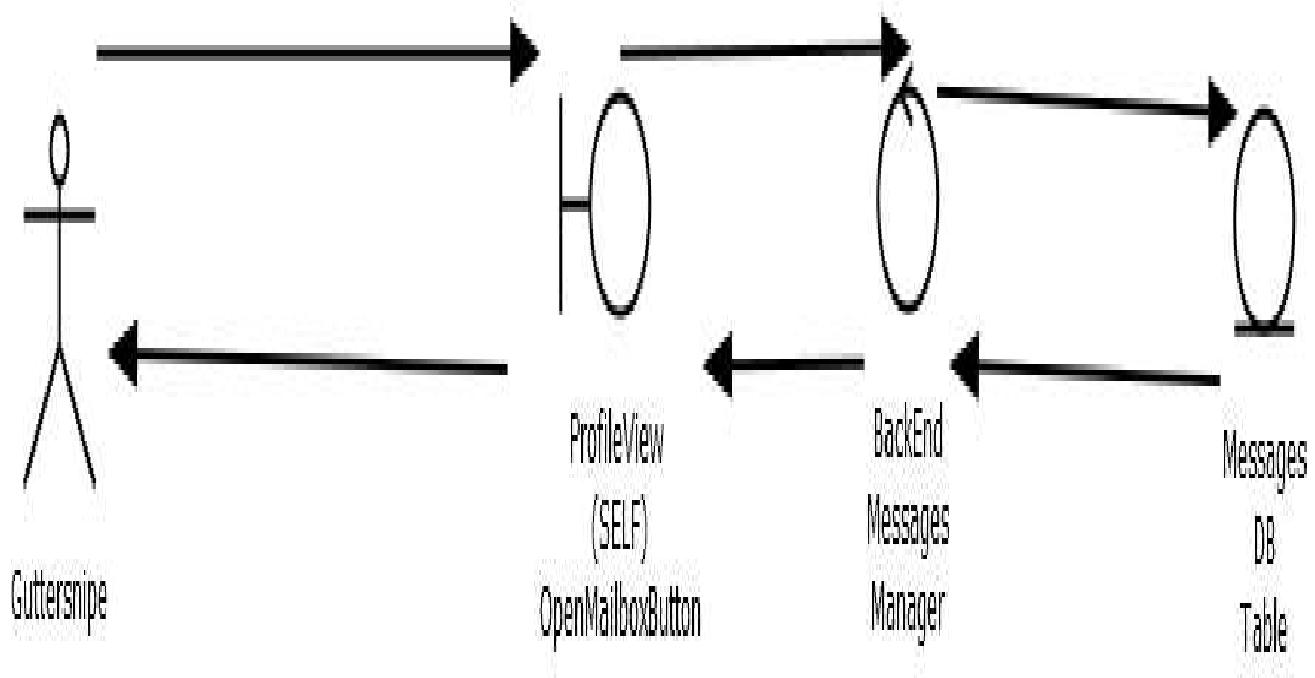
Collaboration Diagram



03.01.04.XX

OpenMailbox

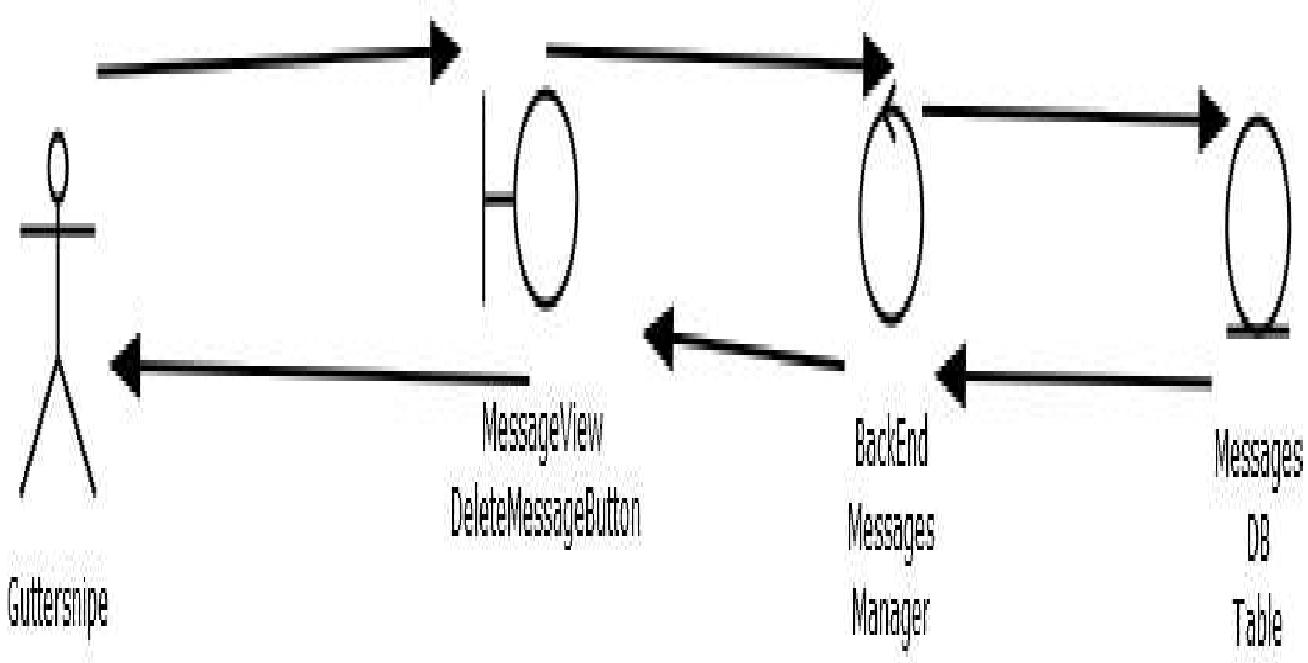
Collaboration Diagram



03.01.04.XX

Delete Message

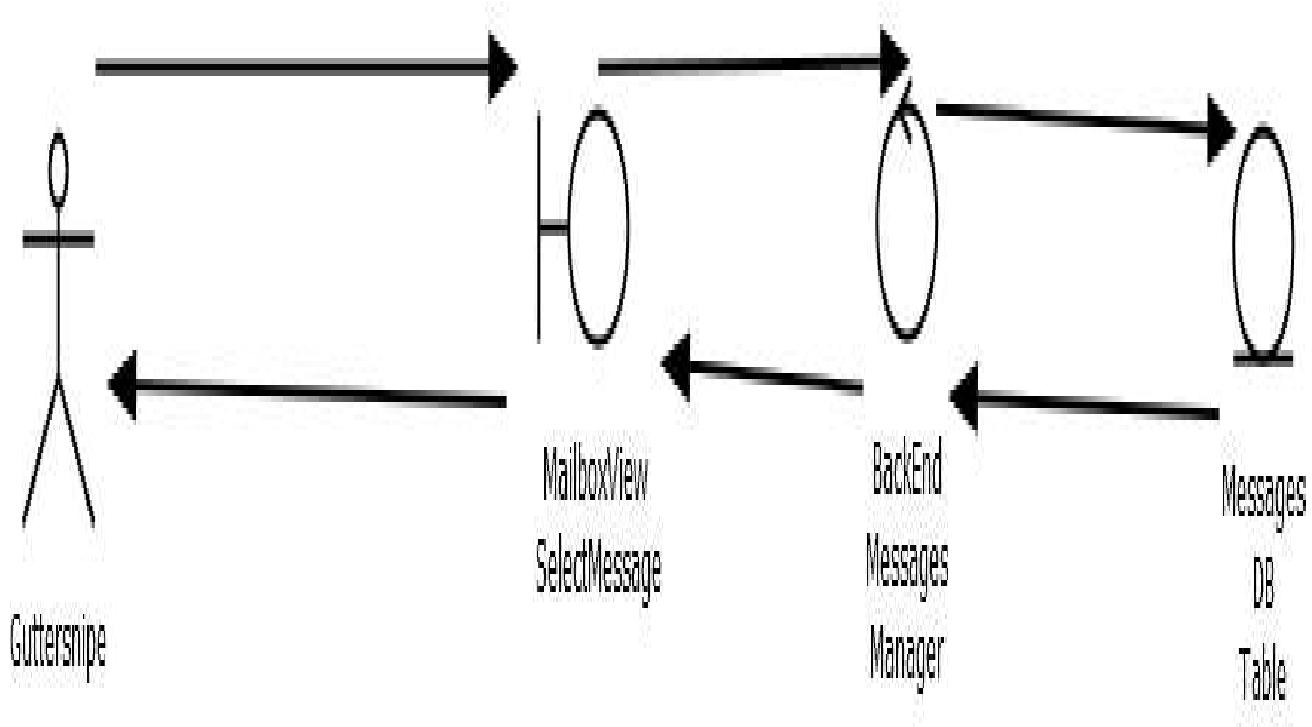
Collaboration Diagram



03.01.04.XX

Read Message

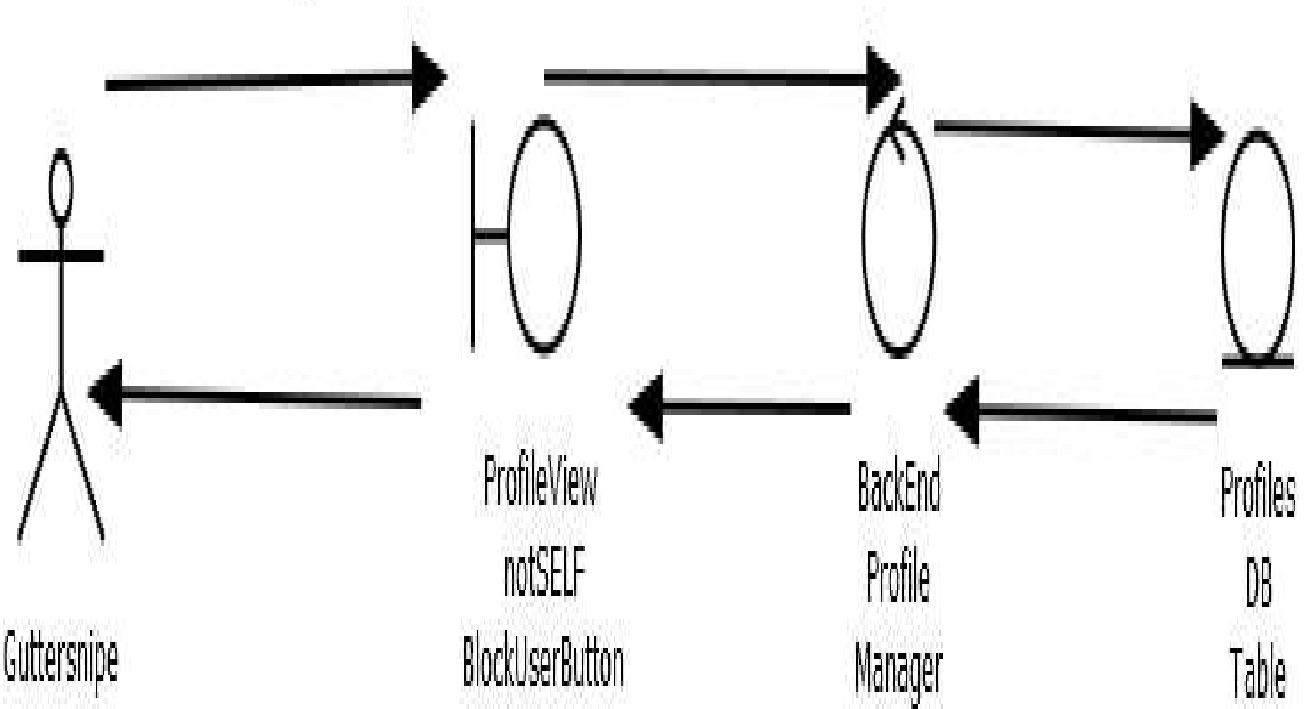
Collaboration Diagram



03.01.04.XX

Block User

Collaboration Diagram

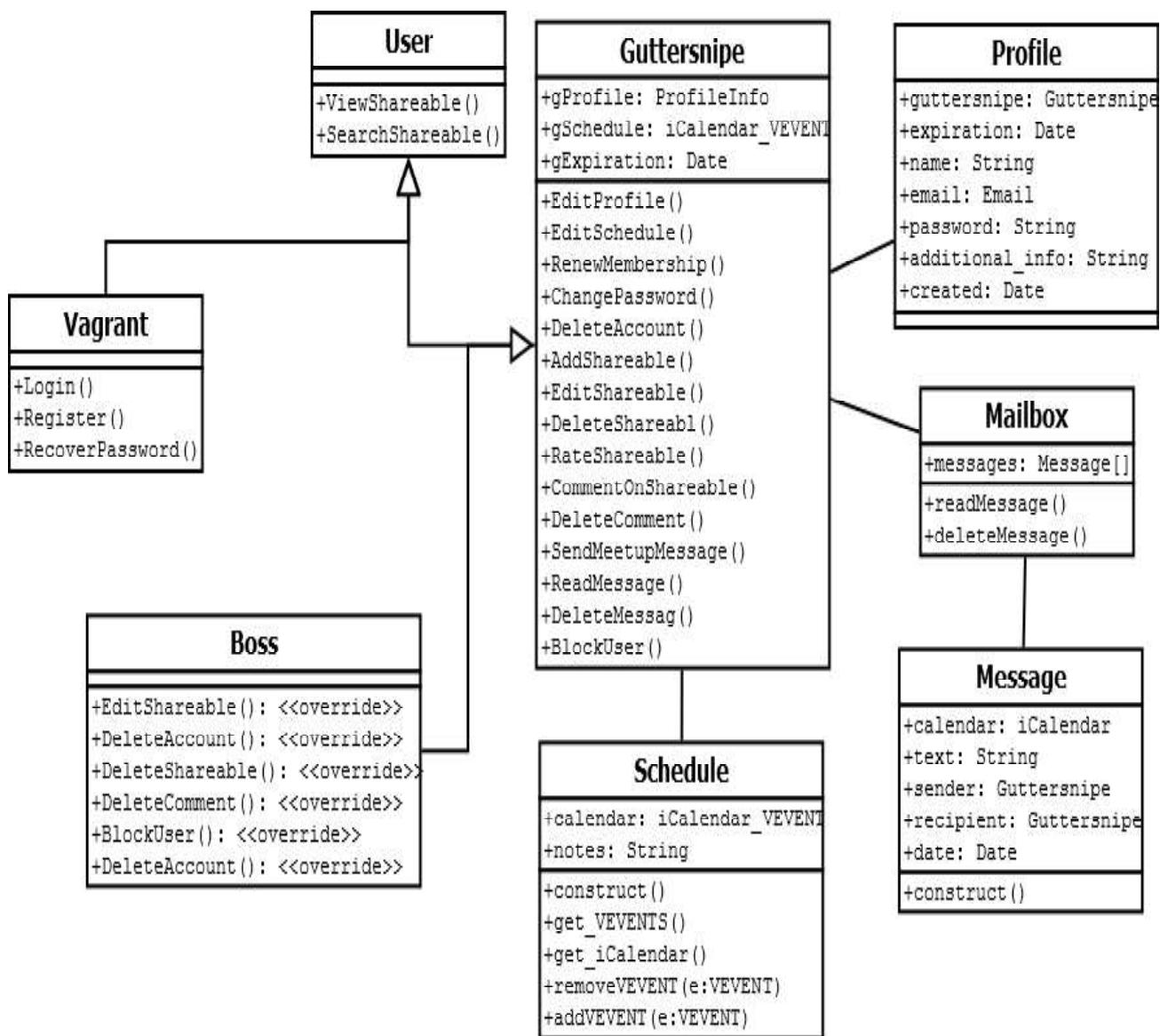


Class D&D

(diagrams && definitions)

User Class

(A) Diagram



03.02.01 User Class Definitions

User Class (abstract)

A User represents anyone using the application. It is an abstract class, so any actual user will be classified as a Vagrant, Guttersnipe, or Boss.

* Methods:

++ViewShareable(): Any User may view a shareable

++SearchShareable(): Any User may search through shareables by selecting filters for Thing, Space, and Time

Vagrant (extends the User Class)

A Vagrant represents someone who is browsing the site and has not yet logged in

* Methods:

++Login(): A Vagrant may log in to its own account

++Register() : A Vagrant may register to become a user

++RecoverUsername: A Vagrant may request to recover its Guttersnipe username

++RecoverPassword: :A Vagrant may request to recover its Guttersnipe username

Guttersnipe (extends the User Class)

A Guttersnipe represents a registered User who has logged into its own Account.

A

*Attributes:

++gProfile: ProfileInfo : Personal Information about the Guttersnipe.

++gSchedule: iCalendar_VEVENT: Availability information for the Guttersnipe

++gLocation: Space. The Guttersnipe's registered location

++gExpiration: Date. A Guttersnipe's account is a Temporary Autonomous Identification, which has an expiration date. This is done for security reasons. The expiration date may be extended at any time.

++ isAdmin: Identifies whether user is Guttersnipe or Boss.

* Methods:

++EditProfile(): A guttersnipe may edit its own profile

++EditSchedule(): A guttersnipe may edit its own schedule

++EditLocation: A Guttersnipe may edit its own location.

++RenewMembership(): A Guttersnipe may renew the termination date of its membership.

++ChangePassword(). A Guttersnipe may change its own password.

++DeleteAccount(): A Guttersnipe may delete its own account.

++AddShareable(): A Guttersnipe may add a Shareable.

++EditShareable(): A Guttersnipe may edit a Shareable that it has added.

++DeleteShareable(): A Guttersnipe may delete a Shareable that it has added.

++RateShareable(): A Guttersnipe may rate any Shareable

++CommentOnShareable(): A Guttersnipe may comment on any Shareable

++DeleteComment (): A Guttersnipe may delete any comment that it has written.

++SendMeetupMessage(): A Guttersnipe may send a message to another Guttersnipe. This message contains schedule and text. It will enable

++ReadMessage(): A Guttersnipe may read a message.

++DeleteMessage(): A Guttersnipe may delete a message.

++BlockUser(): A Guttersnipe may block another

Boss

A Boss is a registered user with administrative capacities. It has the same functionality as a Guttersnipe, but extends several of its methods.

* Methods:

++EditShareable() : A Boss may edit any shareable.

++DeleteAccount(): A Boss may delete any Guttersnipe account

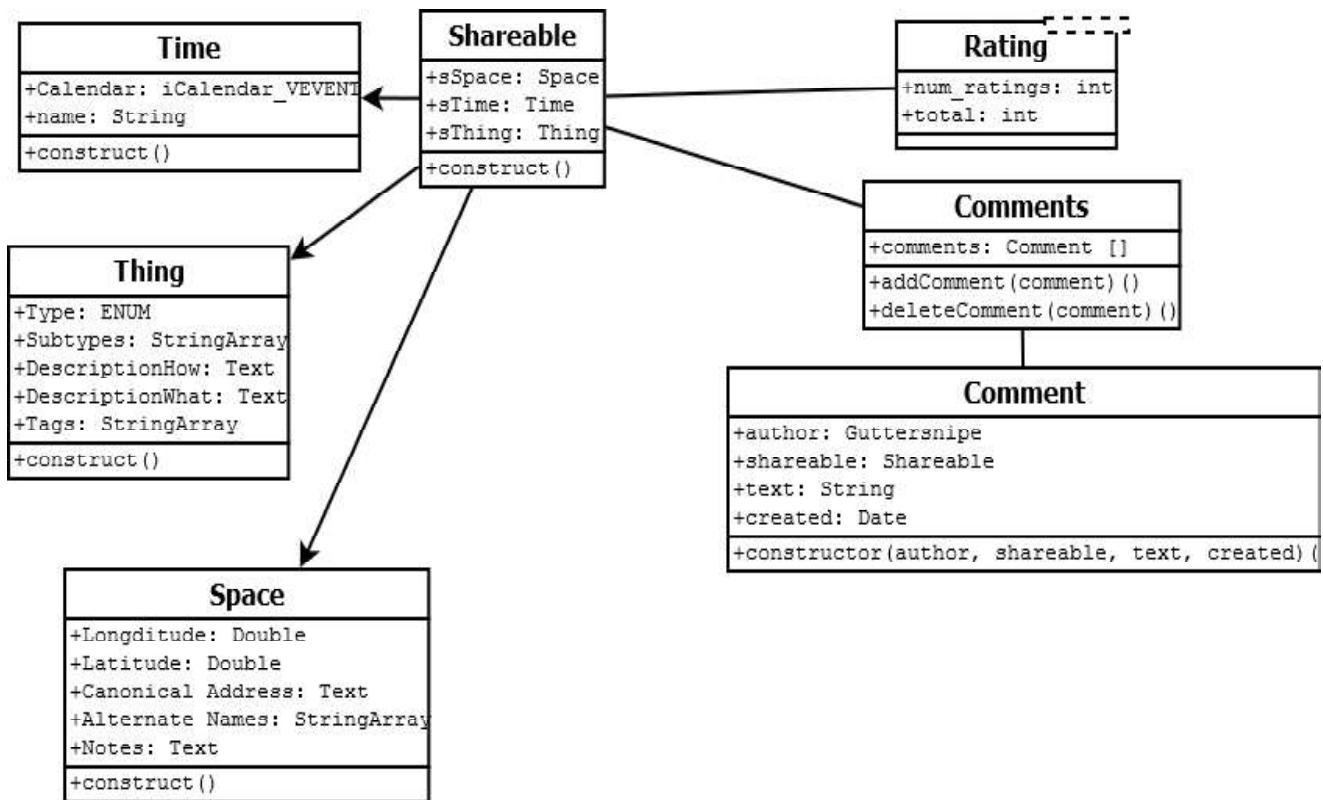
++DeleteShareable () : A Boss may delete any shareable

++DeleteComment () : A Boss may delete any comment

++BlockUser () : When a Boss blocks a Guttersnipe, it may also delete that Guttersnipe's account.

Shareable Class

(A) Diagram



```

Shareable Class and Parts
Shareable
*Attributes:
thing: Thing
space: Space
time: Time
Functions
    construct(thing, space, time) // Constructor for the Shareable class
Thing Class
*Attributes:
type: ENUM
subtypes: String []
descriptionHow: String
descriptionWhat: String
tags: StringArray []
Functions
    construct()
Space Class
*Attributes:
longitude: Double
latitude: Double
canonicalAddress: String
alternateNames: String[]
notes: String
Functions
    construct()
Time Class
*Attributes:
calendar: Calendar_VEVENT
notes: String
Functions
    construct()
Rating Class
Attributes:
num_rating: int
total: int
Functions
    construct()
Comments Class
Attributes:
comments: Comment []
Functions
    construct()
addComment()
deleteComment()
Comment Class
Attributes:
+author: Guttersnip
+shareable: Shareable
+text: String
+created: Date
Functions
*** function __construct($commentID, $commentText, $userName, $timestamp) // Constructor for the Comment class

```

A.

Managers

```

(SearchManager, BrowseManager, MovieViewManager, LoginManager, RegisterManager, MovieRatingsInterface, MovieProcessor, CommentManager, CartManager, CheckoutManager, PasswordControl, LogoutControl, CommentControl, FlaggedCommentsManager, WarningSystem, DeleteManager)

```

i GUI Classes
 (SearchGUI, BrowseGUI, ViewMovieGUI, LoginGUI, MainCustomerPage, MainAdminPage, RegisterGUI, RegistrationSuccessGUI, MoviesBought, MoviePlayerInterface, CommentGUI, ViewCartGUI, CartGUI, CheckoutGUI, ConfirmationGUI, ResetPasswordGUI, GuestPage, FlaggedCommentsGUI, ComplaintInterface, ComplaintsInterface, DeleteUserGUI)

i Database Tables
 (MOVIES, R_USERS, RATINGS, PURCHASES, COMMENTS, COMPLAINTS)

Figure : Complaints.php

Figure :

Figure : register.php

Figure : firstLogin.php

Figure: ViewMovieInfo.php

Figure : browse.php

* Add functionality to current Classes

(
U,
GU, RU, AU, Movie, Cart
)

* Create new Object Classes

(
Player,
Comment,
Session?

All Database Returns Need Object Wrapper

)
* Create new Logic Classes

(
SearchManager, BrowseManager, MovieViewManager, LoginManager,
RegisterManager, MovieRatingsInterface, MovieProcessor, CommentManager,
CartManager, CheckoutManager, PasswordControl, LogoutControl, CommentControl,
FlaggedCommentsManager, WarningSystem, DeleteManager

)
Create new GUI Classes

(
SearchGUI,
BrowseGUI, ViewMovieGUI, LoginGUI, MainCustomerPage, MainAdminPage,
RegisterGUI, RegistrationSuccessGUI, MoviesBought, MoviePlayerInterface,
CommentGUI, ViewCartGUI,
CartGUI, CheckoutGUI, ConfirmationGUI, ResetPasswordGUI, GuestPage,
FlaggedCommentsGUI, ComplaintInterface, ComplaintsInterface, DeleteUserGUI

)
Create Database Tables

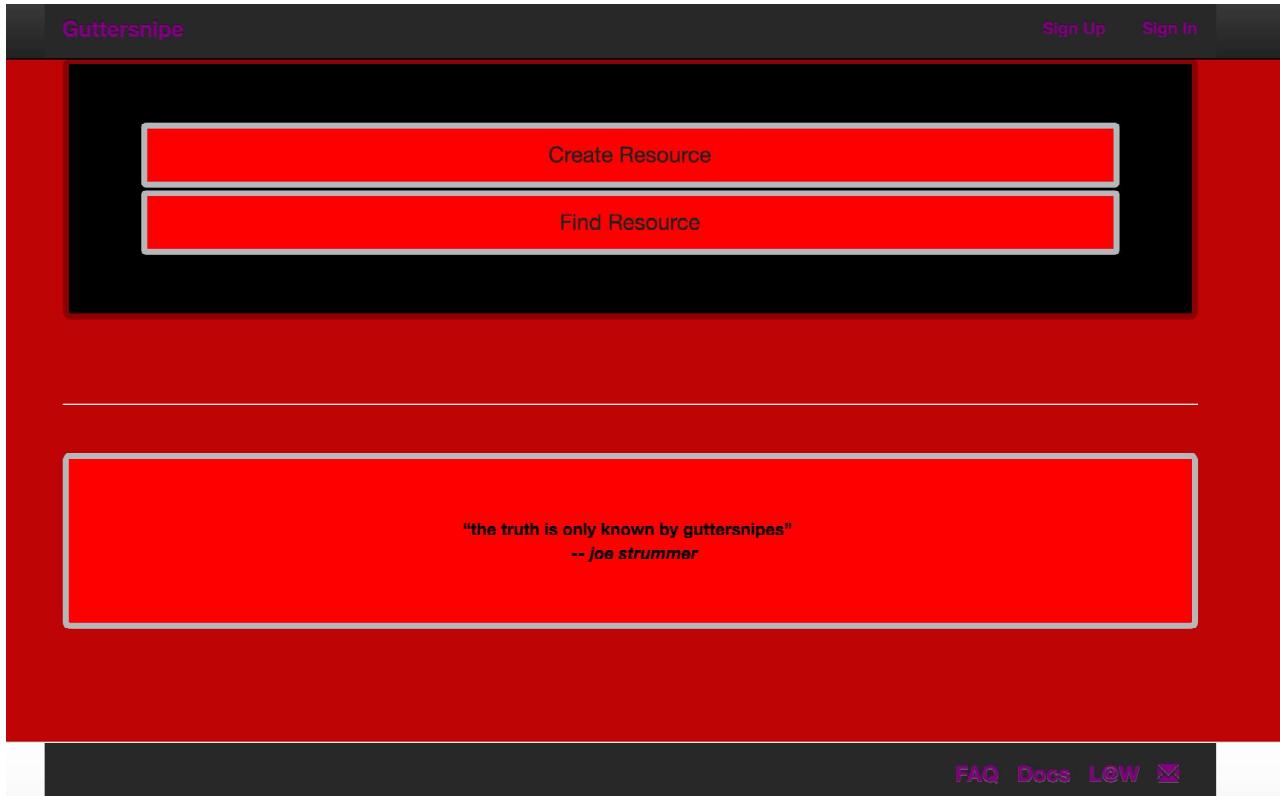
(
MOVIES,
R_USERS, RATINGS, PURCHASES, COMMENTS, COMPLAINTS

GUI COMPONENTS

Screenshots from 0.2 release and Future Wireframes

04.01.01

Front Page:



Front Page shows

- Top Menu Links
 - Home (here)
 - Sign Up
 - Sign In
- Body Links
 - Create Resource
 - Find Resource
 - “the truth is only known by guttersnipes” – joe strummer
- Bottom Menu Links
 - FAQ
 - Documentation
 - Legal
 - Contact

04.02.01

CreateShareable

Start



In order to add a Shareable to the System, the User can enter data in the following Wizard.

This represents the first step of that wizard.

It shows buttons for consent and negation.

Create New Resource

Through your usage of Guttersnipe, you agree to not put yourself or any other person in legal jeopardy.

Negate Consent

You are free to use Guttersnipe as you wish.

04.02.02

Create Shareable Instructions

The screenshot shows a red-themed user interface for creating a new resource. At the top, a blue header bar contains the title 'Create New Resource'. Below this, there are three white input fields with black borders. The first field contains the placeholder text 'Resource = Thing + Place + Time'. The second field contains the example text 'Example: Free meal in Prospect Park every Wednesday from 4PM to 9PM'. The third field contains the instructions: 'To report a Resource on Guttersnipe, please do the following:' followed by a numbered list: 1. Describe Resource, 2. Map Place, 3. Schedule Time, 4. Confirm Resource Report. At the bottom right of the interface is a small blue button labeled 'Create Resource'.

Instructions on how to create Shareable

Create New Resource

Resource = Thing + Place + Time

Example: Free meal in Prospect Park every Wednesday from 4PM to 9PM

To report a Resource on Guttersnipe, please do the following:

Describe Resource

Map Place

Schedule Time

Confirm Resource Report

04.02.03

CreateShareable

Describe

Create New Resource

Describe

Headline

Short explanation

You must provide a headline.

Summary

Resource Description

You must provide a summary.

Method of Access

How do I acquire resource?

Notes

Additional Notes

Confirm Description

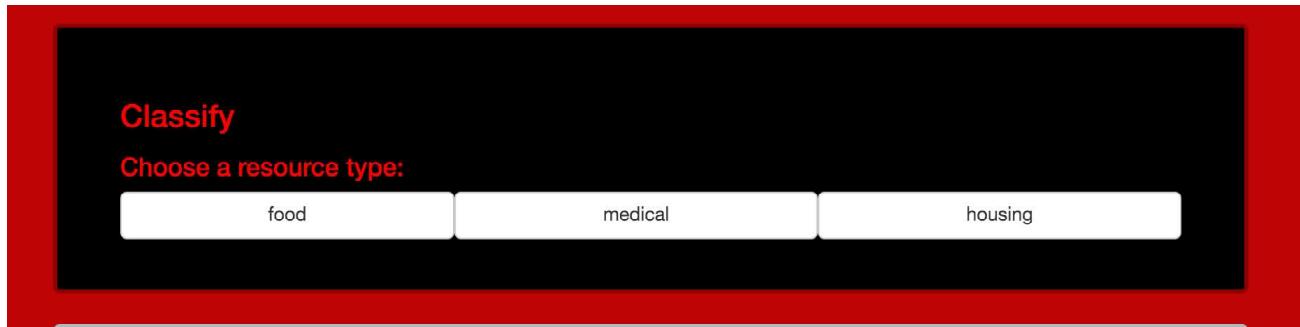
The form is a red rectangle with a white inner area. At the top left is the title 'Create New Resource'. Below it is a section header 'Describe'. Under 'Describe' are four groups: 'Headline' (with a 'Short explanation' input field and an error message 'You must provide a headline.'), 'Summary' (with a 'Resource Description' input field and an error message 'You must provide a summary.'), 'Method of Access' (with a 'How do I acquire resource?' input field), and 'Notes' (with a 'Additional Notes' input field). At the bottom right of the white area is a button labeled 'Confirm Description'.

Form for entering textual data about the Shareable

- Headline
- Summary
- Method of Access
- Additional Notes

04.01.04 (a)

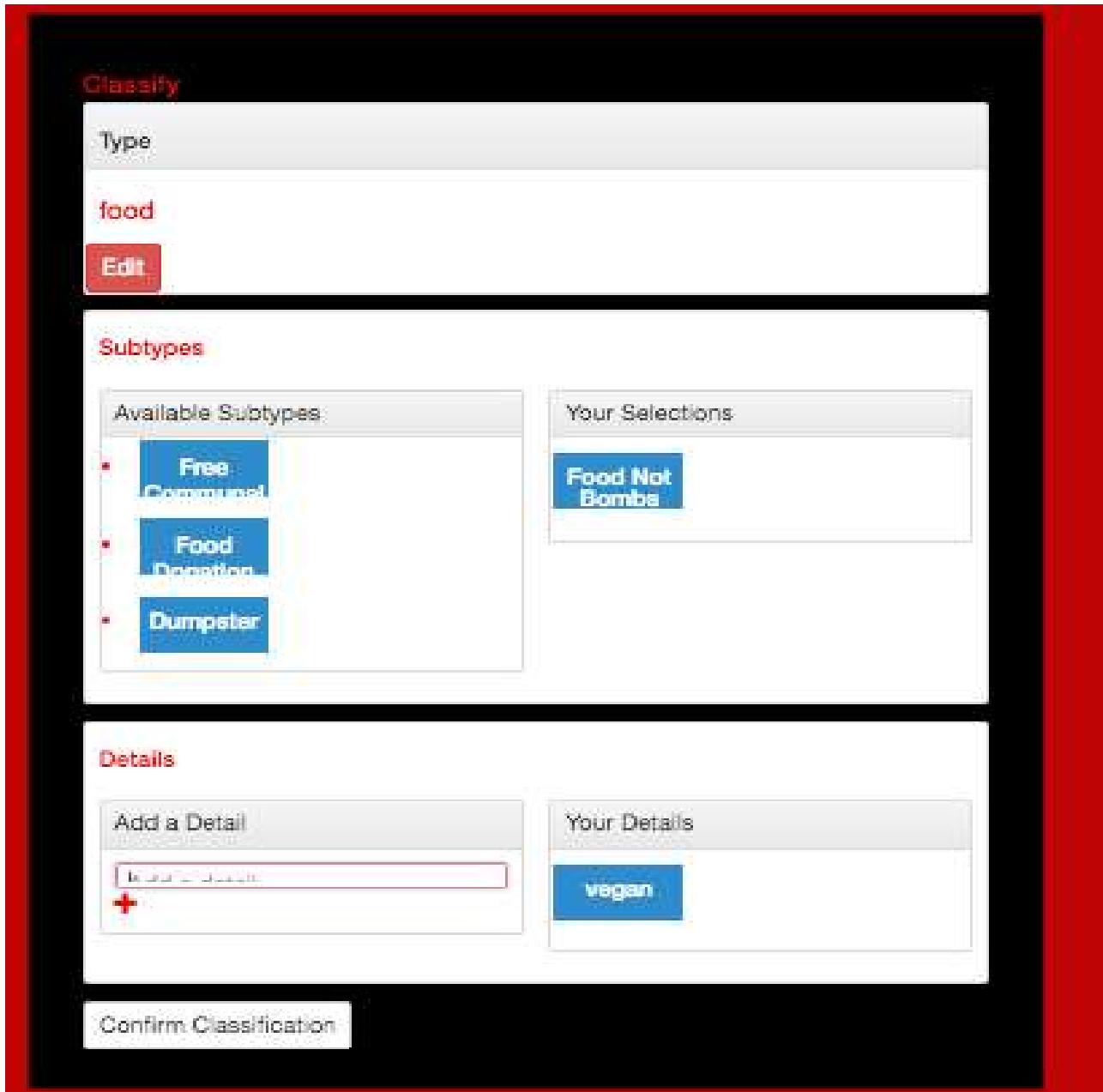
CreateShareable
Classify (1)



Allows Guttersnipe to categorize Shareable as System-defined type

- Food
- Medical
- Housing
- Transport (new)

04.02.04 (b)
Create Shareable
Classify (2)



Allows Guttersnipe to add system-defined **Subtypes** and User-defined **Tags** to Shareable.

04.02.05 (b)

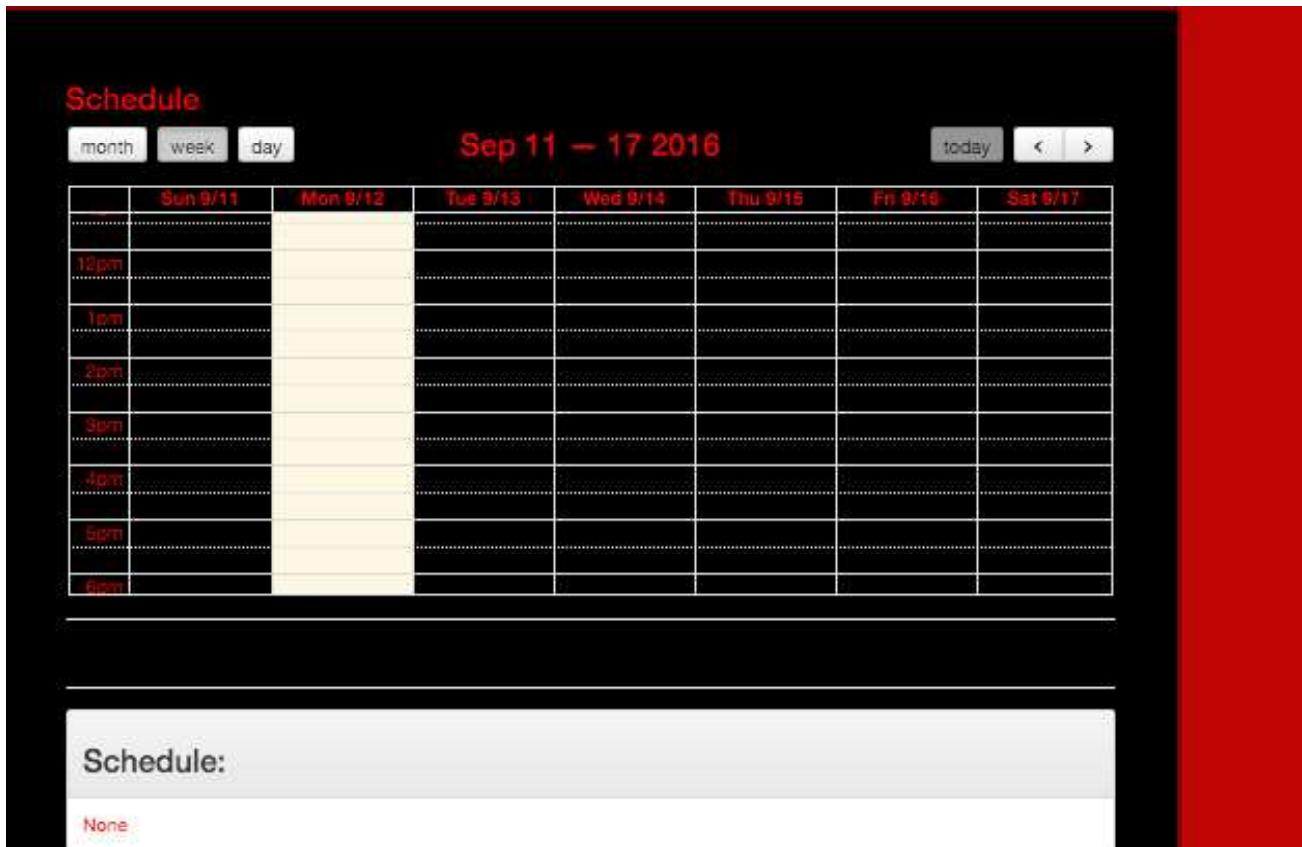
Create Shareable

Map Shareable Location



Shows Map centered on User's current location or Times Square if location is unavailable. Allows user to specify address of the Shareable.

04.02.06(a)
CreateShareable
Schedule (1)

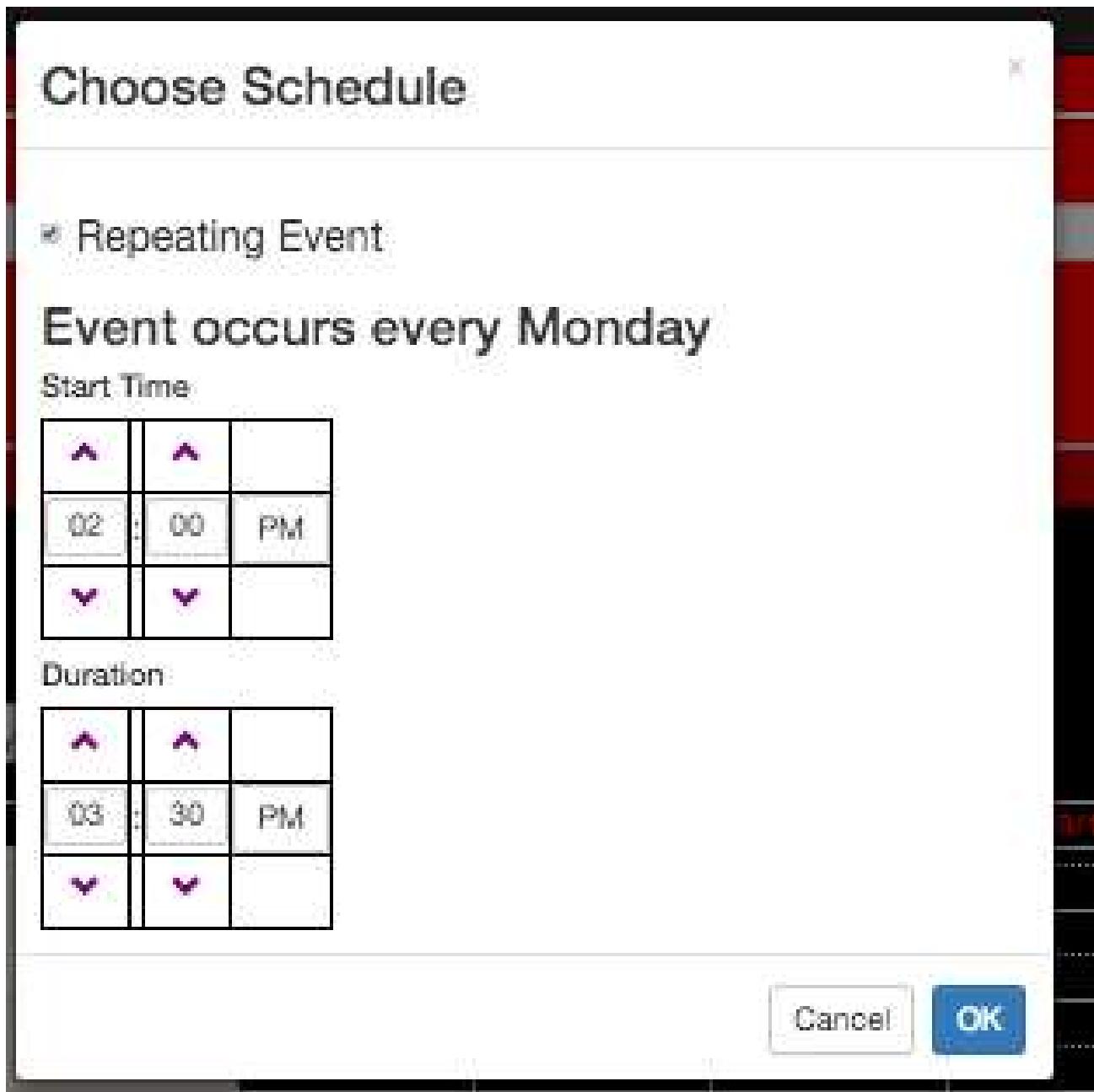


Shows blank calendar. When Guttersnipe clicks on a date, it will be shown popup in next Figure.

04.02.06(a)

CreateShareable:

Schedule (1)



Shows blank calendar. When Guttersnipe clicks on a date, it will be shown popup in next Figure.

04.02.06(c)

Create Shareable Schedule (3)

The image shows a mobile application interface. At the top, there is a red header bar. Below it, the word "Schedule" is displayed in red. Underneath, there are three buttons: "month", "week", and "day". In the center, the text "Sep 11 – 17 2016" is shown in red, with arrows for navigating between weeks. To the right, there are buttons for "today", "<", and ">".

The main area is a grid representing a weekly calendar. The columns are labeled "Sun 9/11", "Mon 9/12", "Tue 9/13", "Wed 9/14", "Thu 9/15", "Fri 9/16", and "Sat 9/17". The rows represent time intervals from "12pm" at the top to "8am" at the bottom. A blue box highlights a specific event: "2:00 - 3:30 Event".

Below this, there is a large white box containing the word "Schedule:" in black. Inside this box, there is a table with the following data:

| Date | Repeats | Start Time | End Time | Delete |
|------|--------------|------------|----------|--------|
| | Every Monday | 02:00 PM | 03:30 PM | X |

At the bottom of the white box, there is a section labeled "Notes" with the text "GET THERE EARLY!!!".

A calendar is shown once the user has selected a new schedule for the event.
A list of schedules is show under the calendar.
The user can delete any schedules which are incorrect.

04.03. SearchShareable

04.03.01

SearchShareable Results List

The screenshot shows a search results page titled "Resources". It displays three items, each with a "Full Record" button:

- La Bagel Delight**
 - Type: food
 - Subtypes: dumpster

104 Front St
Brooklyn, NY
11201

 - Every Monday
From Monday, 12:00 AM
To Monday, 2:59 AM
 - Every Tuesday
From Tuesday, 12:00 AM
To Tuesday, 2:59 AM
- Tader Joe's**
 - Type: food
 - Subtypes: dumpster

130 Court St
Brooklyn, NY
11201
- Peregrina**
 - Type: food
 - Subtypes: dumpster

175 Remsen St
Brooklyn, NY
11201

 - Every Monday
From Monday, 12:45 AM
To Monday, 2:59 AM

Created by Apr 16, 2015 1:21:24 PM by

Filter Resources by:

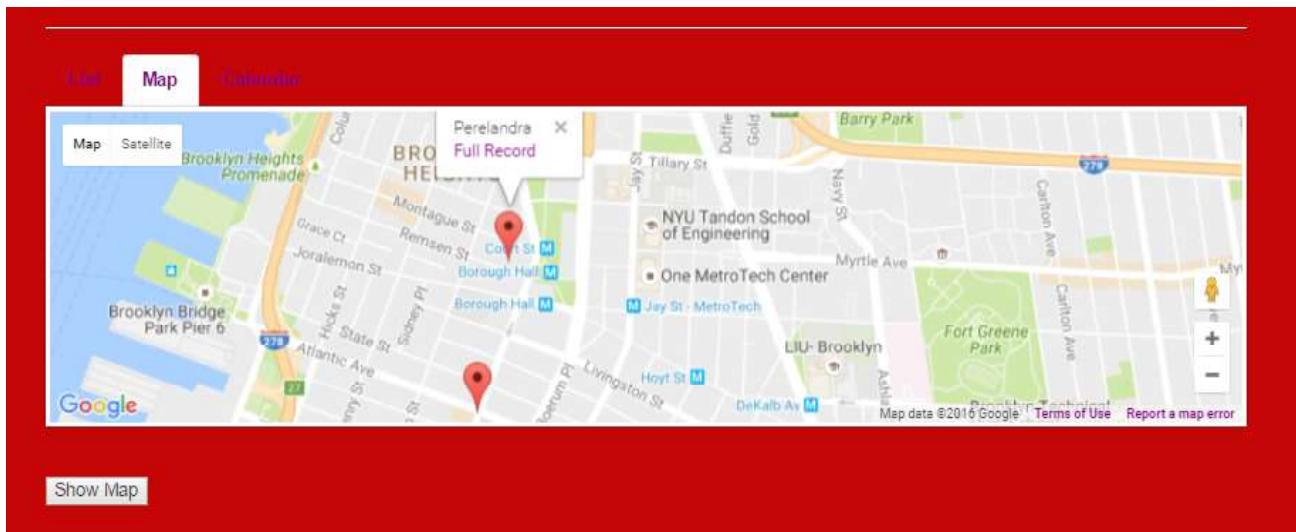
Thing People Other

Shows the results of the Shareable Search as a list.
Columns for

- Headline
- Thing (Type && Subtype)
- Place
- Times
- Link to Full Record

04.03.02

SearchShareable Results Map



Shows Shareable Search Results as a Map.

Each point on the map has a tooltip that opens up a window with a Summary and a link to the Shareable.

04.03.03

SearchShareable Results Calendar

The screenshot shows a calendar interface for September 2016. The top navigation bar includes tabs for 'List', 'Map', and 'Calendar', with 'Calendar' being the active tab. Below the tabs is a date header 'September 2016'. To the right of the date are buttons for 'today', '<', and '>'. The main area is a grid representing the month. Each day of the week has a header row with columns for 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', and 'Sat'. The days themselves are represented by blue boxes containing search results. For example, Monday, September 5th, contains two entries: 'La Bagel Delight' and 'Perelandra'. The grid spans from September 1st to September 30th. At the bottom left is a 'Show Calendar' button, and at the bottom center is a 'Filter Resources by' input field.

Shows Shareable Search results as a Calendar.
Each entry on the calendar a link that can be clicked for the full record.

04.03.03

SearchShareable Search Filters



Search Interface will have 3 filters. These have not been designed yet

- Thing
 - By Type and Subtype
 - By Tags
- Time
 - By Date/Time
- Space
 - Choose Map Location

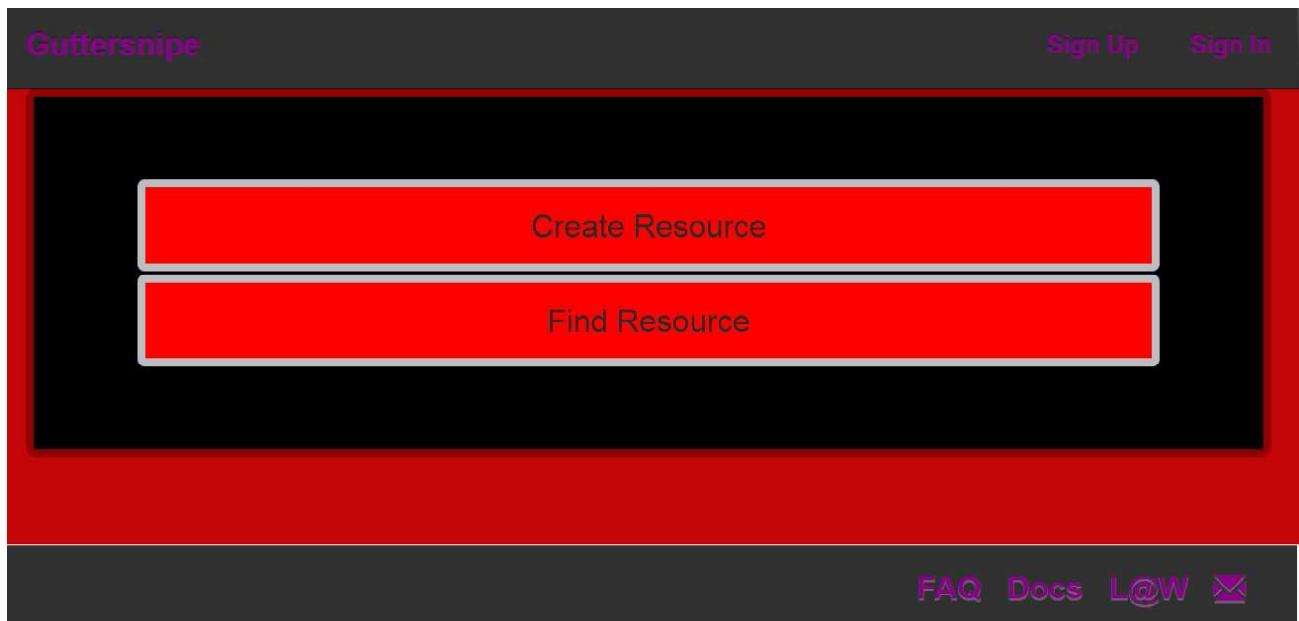
User will be able to refine these 3 search filters and submit a search query when done.

04.03. Account Management

04.04.01

Account Management

Sign In and Sign Up Buttons



In the top right corner of the menu bar are buttons for “Sign Up” and “Sign In”

04.04.02

Account Management

Smaller Screens



On Smaller Screens, users can click on the top right burger-shaped menu button to display the Sign Up and Sign In buttons

04.04.05

Account Management

Sign In

The screenshot shows a mobile application interface for 'Guttersnipe'. At the top, there is a dark header bar with the 'Guttersnipe' logo on the left and a three-line menu icon on the right. Below the header is a red background area containing a sign-in form. The form includes a 'Username' label and a white input field. Below that is a 'Password' label and another white input field. To the right of the password field is a blue button labeled 'Sign in'. To the right of the 'Sign in' button is the word 'or' followed by a link 'Sign up'. At the bottom of the red area, there is a link 'Forgot your password?'. At the very bottom of the screen is a dark footer bar with links for 'FAQ', 'Docs', 'L@W' (with a mail icon), and another mail icon.

Shows form

- User
- Password
- Sign In Button

04.04.04

Account Management

Sign Up

Sign up

First Name

Last Name

Email

Username

Password

Sign up or Sign in

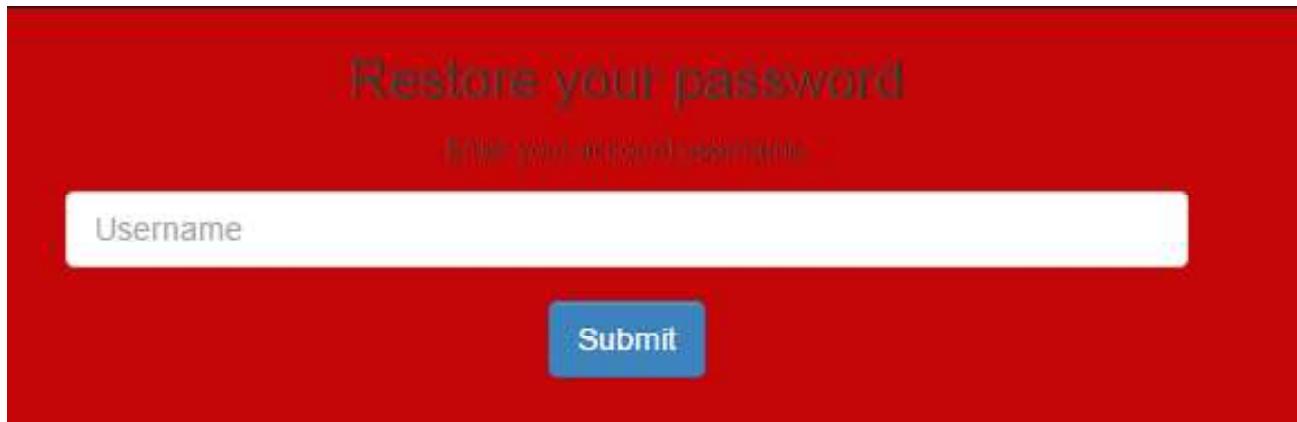
Signup Page asks user to fill in form:

- Username:
- Expiration Date
- Email (optional)
- Password (optional)

04.04.05

Account Management

Restore Password



Provides a button for asking system to send user a password reset form.

Documentation

DOCUMENTATION: GENERAL

04.05.01

Documentation: General Mission Page

The screenshot shows a website's mission page. At the top, there is a red header with white text that reads "FOR THE KIDS!!!". Below the header, the main content area has a red background. A quote by Peter Kropotkin is displayed in white text:

By what means would a city in a state of revolution be supplied with food? We shall answer this question, but it is obvious that the means resorted to will depend on the character of the Revolution in the provinces, and in neighbouring countries. If the entire nation, or, better still, if all Europe should accomplish the Social Revolution simultaneously, and start with thorough-going Communism, our procedure would be simplified; but if only a few countries in Europe make the attempt, other means will have to be chosen. The circumstances will dictate the measures.

Below the quote, there is a section titled "The Conquest of Bread by Peter Kropotkin" with a "Read More" button. At the bottom of the page, there is a video player showing a clip of Joe Strummer singing "Guttersnipe" from The Clash. The video player has a play button and a progress bar.

Mission page shows

- “FOR THE KIDS!!!”
- Kropotkin Quotes Widget
- Joe Strummer “guttersnipe” video.

04.05.02

Documentation: General FAQ Page

The screenshot shows a document titled "PROPOSAL - GUTTERSNIPE FAQ". The page contains a list of frequently asked questions, each with a question number, a question title, and a detailed answer. The questions are numbered 1 through 10. The answers are long and contain several lines of text, some of which are partially obscured by red bars.

- 1. What is the guttersnipe?
- 2. What kind of project is this?
- 3. What is the purpose of the project?
- 4. Who is funding this project?
- 5. Who is leading this project?
- 6. What is the goal of this project?
- 7. Why did they choose this platform?
- 8. Why will they keep coming back to your app?
- 9. How many different forms will there be?
- 10. What's the timeline?

Displays a FAQ about the project.

This is same content as section “01_01: Proposal” of the current document.

DOCUMENTATION:

2013 Presentation

04.05.03.01

Documentation: 2013 Presentation
Start Page



From 2013 Presentation:

Section 6:

START PAGE

The following slides are taken from my 2013 presentation prepared for what I intended to be my senior project at City College of New York

04.05.03.02

Documentation: 2013 Presentation Front Page

The screenshot shows a presentation slide with a red header bar containing navigation links: 1 Start, 2 Objective, 3 Audience, 4 Consultations, 5 Other Sites, 6 Q:Method, 6 A:Method, 6 Example, 6 Report, 6 Result, 6 Research, and 6 Fin. Below the header is a large black rectangular area containing the presentation content. The content includes the title 'GUTTERSNIPE' in a stylized font, the URL 'http://www.guttersnipe.org', the author's name 'Mitchell Verter', a small logo consisting of three icons (a roadrunner, a gear, and a hand), the email address 'roadrunner [at] waste [dot] org', the date 'March 4, 2013', the text 'Presentation for Cs59866, Senior Project', 'Professor Izidor Gertner', a portrait photo of Alexander Berkman, and a small caption in Hebrew: 'הנשען על מילון מילון'. At the bottom left is a 'NEXT' button.

From 2013 Presentation:

Section 6:
FRONT PAGE

=====

GUTTERSNIPE

<http://www.guttersnipe.org>

Mitchell Verter

roadrunner [at] waste [dot] org

March 4, 2013

Presentation for Cs59866, Senior Project

Professor Izidor Gertner

<Picture of Alexander Berkman>

04.05.03.03

Documentation: 2013 Presentation Objective

Presentation: 2013

1 Start 2 Objective 3 Audience 4 Consultations 5 Other Sites 6 Q:Method 6 A:Method 6 Example 6 Report 6 Result 6 Research 6 Fin

Objective

- To overthrow capitalism by helping to establish mediums of exchange outside of the capitalist marketplace.
- "Over a billion human beings live in absolute poverty, suffering from chronic malnutrition and other ills, while we have much more than an adequate material basis for a good life for all." – John Clark, *The Impossible Community Realizing Communitarian Anarchism*

PREV NEXT

From 2013 Presentation:

Section 6: OBJECTIVE

=====

- To overthrow capitalism by helping to establish mediums of exchange outside of the capitalist marketplace.
- "Over a billion human beings live in absolute poverty, suffering from chronic malnutrition and other ills, while we have much more than an adequate material basis for a good life for all." – John Clark, *The Impossible Community Realizing Communitarian Anarchism*

04.05.03.04

Documentation: 2013 Presentation Audience

The screenshot shows a presentation slide with a red header bar containing navigation links. The main content area has a black background with white text. The title 'Audience' is at the top, followed by a bulleted list of people who cannot afford or choose not to participate in the capitalist marketplace. At the bottom of the slide are 'PREV' and 'NEXT' buttons.

Presentation: 2013

1 Start 2 Objective 3 Audience 4 Consultations 5 Other_Sites 6 Q:Method 6 A:Method 6 Example 6 Report 6
Result 6 Research 6 Fin

Audience

- People who can not afford to or who choose not to participate in the capitalist marketplace
 - The Victimease Majority
 - Poor
 - Dislocated
 - Immigrants
 - Street people
 - Homeless
 - Traveler punks
 - Contingent/Precarious workers
 - Lumpenproletariat

PREV NEXT

From 2013 Presentation:
Section 6:

AUDIENCE

=====

People who can not afford to or who choose not to participate in the capitalist marketplace

- The Victimease Majority
- Poor
- Dislocated
- Immigrants
- Street people
- Homeless
- Traveler punks
- Contingent/Precarious workers
- Lumpenproletariat

04.05.03.05

Documentation: 2013 Presentation
Consulting Organizations

The screenshot shows a presentation slide with a red header bar. The header bar contains the text "Presentation 2013" and a navigation menu with the following items: 1 Start, 2 Objective, 3 Audience, 4 Consultations, 5 Other Sites, 6 Q:Method, 6 A:Method, 6 Example, 6 Report, 6 Result, 6 Research, and 6 Fin.

Consulting Organizations

- **Freegan NYC:**
 - Freegans are people who employ alternative strategies for living based on limited participation in the conventional economy and minimal consumption of resources.
 - Dumpster Dive Directory.
- **Picture the Homeless**
 - A grassroots organization founded and led by homeless people.
 - Extensive list of vacant properties.

PREV NEXT

From 2013 Presentation:

Section 6:

CONSULTING ORGANIZATIONS

Consulting Organizations

- **Freegan NYC:**
 - Freegans are people who employ alternative strategies for living based on limited participation in the conventional economy and minimal consumption of resources.
 - Dumpster Dive Directory
- **Picture the Homeless**
 - A grassroots organization founded and led by homeless people.
 - Extensive list of vacant properties

04.05.03.06

Documentation: 2013 Presentation Other Sites



From 2013 Presentation:

Section 6: OTHER SITES

=====

Other Web Sites

- Other Websites
 - Squat.net
 - Foodsharing (Germany)

04.05.03.07

Documentation: 2013 Presentation Question of Method



From 2013 Presentation:

Section 6:

QUESTION OF METHOD

=====

Question of Method

- . How do you get to Carnegie Hall?

04.05.03.08

Documentation: 2013 Presentation Answer of Method

Presentation: 2013

1 Start 2 Objective 3 Audience 4 Consultations 5 Other Sites 6 Q:Method 6 A:Method 6 Example 6 Report 6
Result 6 Research 6 Fin

Answer of Method

Answer: Geolocation

• 881 7th Ave
New York, NY 10019



PREV NEXT

From 2013 Presentation:
Section 6:

ANSWER OF METHOD

=====

Answer: Geolocation

881 7th Ave
New York, NY 10019

04.05.03.09

Documentation: 2013 Presentation Example (2002)



The following pages are taken from a previous incarnation of this project.
Designed in 2002, these pages are still available at:
<https://www.waste.org/~roadrunner/squat/>

04.05.03.12

Documentation: 2013 Presentation Submit Property Report GUI (2002)

The screenshot shows a web-based application interface for submitting a property report. At the top, there is a navigation bar with links: Home & Objective, Audience & Consultations, Other Ideas, Method & A-METHOD, Example & Report, Read & Research, and Print.

The main title is "Submit a property report". Below it, there is a brief description: "Submitting your report information is very easy. To example, if you want to report about a building, address, telephone number or under name, just enter the address, street, city, state, zip code and under name." There is also a link to "View instructions" for more details.

Below the description, there is a section titled "Street Address:" with three input fields: "Number", "Street", and "Type". The "Number" field contains "1234", the "Street" field contains "Main Street", and the "Type" dropdown menu is set to "Alley".

Next is a "Zip Code:" section with a single input field containing "12345".

Then comes a "Report" section with a large text area for entering the report content. A placeholder text "Please describe what you have found out about the subject." is visible inside the text area.

At the bottom, there is a "Disclaimer" section with a long, mostly illegible text block. It includes several checkboxes and radio buttons, some of which are checked. The text discusses the limitations of the service and the user's responsibility to verify information.

Shows the Submit Property Report interface from the 2002 deployment.

04.05.03.11

Documentation: 2013 Presentation Property Submission Response (2002)

INFORMATION 2013

1 Start 2 Objective 3 Audience 4 Consultations 5 Other Sites 6 Q:Method 6 A:Method 6 Example 6 Report 6
Result 6 Research 6 Fin

Result



Map Satellite Hybrid

Map data ©2013 Google - [Terms of Use](#)

Property report submitted

You have submitted the following information for the address

73 st pauls pl
11226

shorty The Dog's place.

[PREV](#) [NEXT](#)

Shows a Resulting page for a Property Submission from the 2002 deployment.

04.05.03.12

Documentation: 2013 Presentation Property Search (2002)

The screenshot shows a web-based application titled "Find vacant properties". The interface includes a form for entering address information, a "Report" section, and various navigation links.

Find vacant properties

Use the following form to research a vacant property and its neighborhood.

For example, to research 1015 Polk Street in San Francisco, California, 94109, type in "1015" under Number, type in "Polk" under Name, choose "Street" under Suffix, type in "94109" under Zip Code.

You must fill out all fields. If you are having difficulty determining the zip code, you should try looking at some [Google Maps](#).

Then describe what you have found under Report.

Street Address:

| Number | Name | Suffix |
|--------|------|--------|
| | | Alley |

Zip Code:

Are you having problems determining the address? Maybe you should try looking at some [Google Maps](#).

Report

[Find Properties](#) [Reset](#)

[W.R. SITE search.html](#)

[Search a vacant property](#)

[Look at maps](#)

[Read Xanadu's thoughts on this project](#)

[PREV](#) [NEXT](#)

Shows the Property Search GUI from the 2002 deployment.

04.05.03.13

Documentation: 2013 Presentation
FIN



Final Page from 2013 Presentation:

For Joe Strummer
“The truth is only known by guttersnipes”

<http://www.guttersnipe.org>

Mitchell Verter

roadrunner [at] waste [dot] org

DOCUMENTATION: ADMINISTRATIVE INFORMATION PAGES

04.05.01

Documentation: Administrative L@W Page



Shows Legal Information

=====

You are free to use Guttersnipe as you wish.

All Wrongs Righted
All Rites Reversed

GNU General Public License

04.05.04.02

Documentation: Administrative Credits Page



Shows creator and inspirations for the project.

04.05.04.03

Documentation: Administrative
Contact Information



Displays email address and a clickable picture to send mail.

Kropotkin Quote Widget

04.06.01

Kropotkin Quote

By what means could a city in a state of revolution be supplied with food? We shall answer this question, but it is obvious that the means resorted to will depend on the character of the Revolution in the provinces and in neighbouring countries. If the entire nation, or better still, if all Europe should accomplish the Social Revolution simultaneously and start with thorough-going communism, our procedure would be simplified. But if only a few communities in Europe make the attempt, other means will have to be chosen. The circumstances will dictate the measures.

The Conquest of Bread by Pyotr Alexeivitch Kropotkin

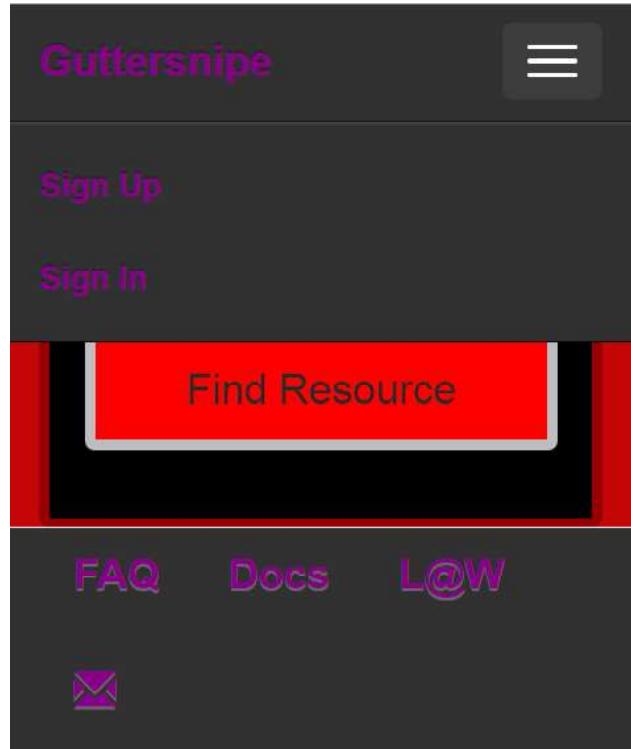
New Quote

The Kropotkin widget displays quotes from “Conquest of Bread”
The “New Quote” button produces another randomly-chosen quote.
We intend to add another button that cycles through quotes every minute.

RESPONSIVE WEB DESIGN

04.04.07

Menu Dropdown



On smaller screens, the top toolbar is replaced with a button which displays a dropdown menu which reveals, when pressed, buttons for Sign In and Sign Up