Software Requirements Specification

Version-2.0

9th March, 2014

IOU Tracking with Friends

Team 2

Soumen Naskar(M120366CA) - Leader

Meenakshi Rani(M120392CA)

Nagendra Negi(M120540CA)

Puneetha N(M120396CA)

Submitted in partial fulfillment
Of the requirements of

CS 3004 Software Engineering

Table of Contents

1.0.Introduction:

1.1 Purpose:

The purpose of this document is to present a detailed description of all the functions and specifications of the IOU online tracking with friend's application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.2 Scope:

IOU application will be designed to help the users in order to track informal debts, expenses, and bills, by allowing users to enter notes about who they owe, who owes them and why. It will allow the users to create groups, add friends, add expenses and accept or decline debts. The Service keeps track of the sharing math, provides informal sharing advice, and stores this data for later use. With IOU system, one will not forget who owes him/her how much.

1.3 Glossary:

Term	**Definition**
IOU	I Owe You
User	All the members of the group registered in IOU are considered here as user.
Group leader	The person who creates the group will be called as group leader.
Members	All the persons who are being added in the group.
Group	Friends who are sharing their informal debts.
Account	Account here refers to the person's profile. Information about the total balance which includes the amount to be paid/paid by others, etc.
Transaction chart	Total summary (debts, refunds, etc) of all the members in a group. Transaction nowhere here refers cash transactions.

1.4 References:

[1] IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

[2] Software Requirements Specification Web Publishing System

1.5 Overview of the document:

The remainder of this document is two chapters, the first providing a full description of the project for the owners/users of the IOU. It lists all the functions performed by the system. The final chapter concerns details of each of the system functions and actions in full for the software developers' assistance. Both sections of the document describe the same software product in its entirety, but are intended for different audiences.

2.0 Overall Description:

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions of the system will be presented.

2.1 Product Perspective:

IOU is a web-based totally independent and self-contained software. The IOU system contains a database which holds all the information about the users. The application keeps track of all the personal debts of a registered group. It also facilitates communication around who owes who, and for what. The repaid debts history will also be saved. The system will be completely web-based. An Internet connection is necessary to access the system.

2.1.1 System Interfaces:

A web server will be used. The user inputs data via the web server using HTML forms. The actual program that will perform the operations is written in PHP.

2.1.2 User Interfaces:

Login Interface: The registered user will sign in by giving the required credentials. The new users have to sign up to get into the IOU system.

Main Interface: After login or sign up the user will be in his/her home profile where the user will be shown the present state of his account, notifications, the groups where the user belong to. He can navigate to see all of his expenses that includes the amount that the person owes or owed, all the transactions in a tabulated form, can add an expense, settle up his/ her account, pending bills etc. By selecting any group the user can see all the information including the transactions in the group, the status of the members, and can change the group settings but for that the person have to have necessary privilege. From user home user can edit his information, can add friends, can create groups etc.

2.1.3 Hardware Interfaces:

a) Server side

The web application will be hosted on a web server which is listening on the web standard, port 80.

b) Client side

- Display Monitor
- Keyboard
- Mouse
- Internet connection.

2.1.4 Software Interfaces:

- HTML
- MySQL
- Javascript/PHP

2.1.5 Communication Interfaces:

The HTTP protocol will be used to facilitate communication between the client and server.

2.1.6 Memory constraints:

Since this will be a web-based application, the main concern is on the internet speed.

2.2 Product Functions:

2.2.1 System Environment:

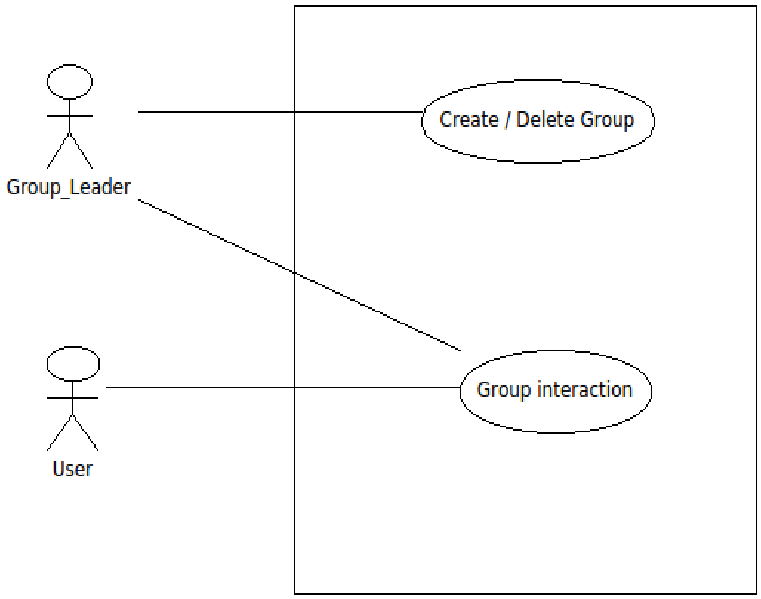


Figure1. System Environment

The I Owe You system consists of as many active users at the same time that belongs to a particular group. Any member can create a group, which will become the group leader and add their friends. Members can access this application through Internet.

2.3 Functional Requirement Specification:

This section highlights the use cases for each users. In this application, all the members of the group are considered as users. A group leader who creates the group will act as a user unless there is some need to delete the group i.e. There will be a special privilege to a particular user who creates the group. He/she will have the authority to delete the group after getting approval from the group members. Whereas members can leave the group any time they prefer.

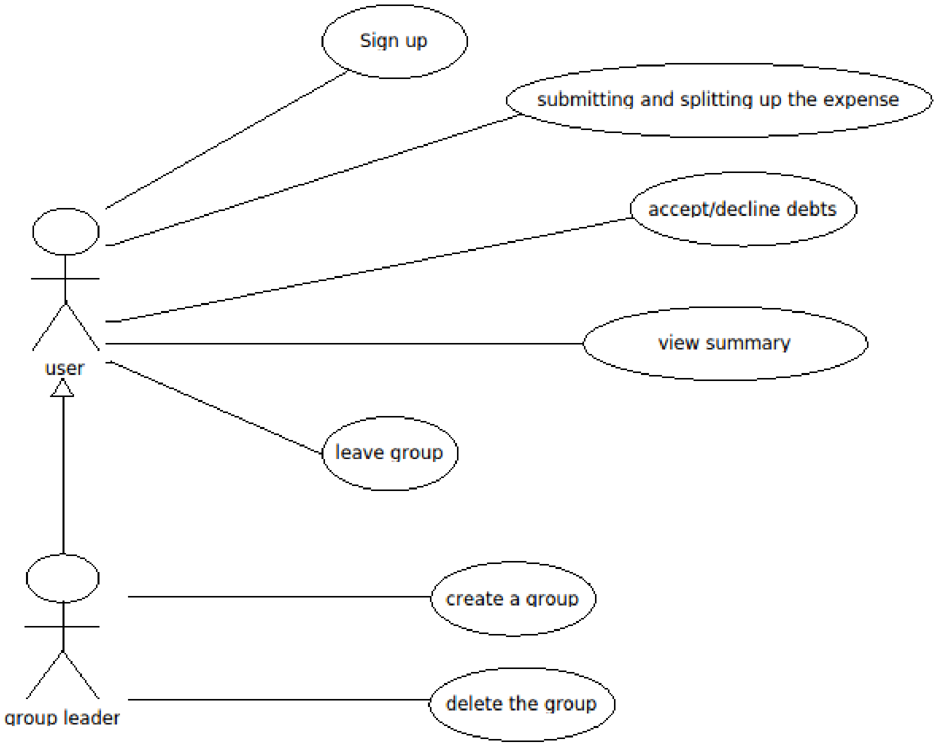
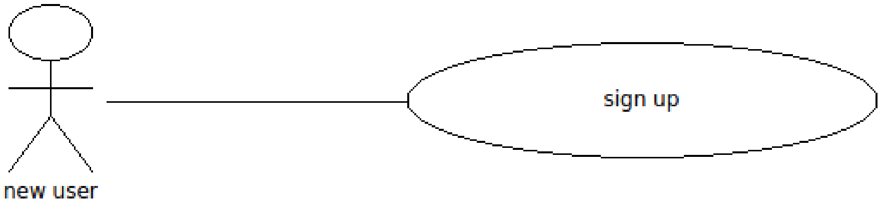


Figure 2. Use Case diagram

2.3.1 Use Case: Sign up

Diagram:



Brief description:

The new user will register into the IOU.

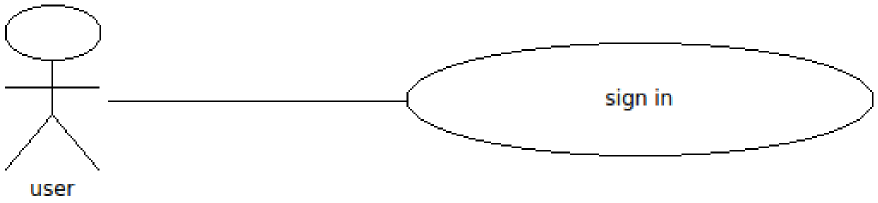
Initial step-by-step description:

1. The home page provides the facility to register.
2. The new user will register with a unique username and valid password.
3. The registered user will enter into the application.

Xref: Section 3.1.1,Sign up

2.3.2 Use case: Sign In

Diagram:



Brief Description:

The user will sign in to the system after signing up to make use of IOU.

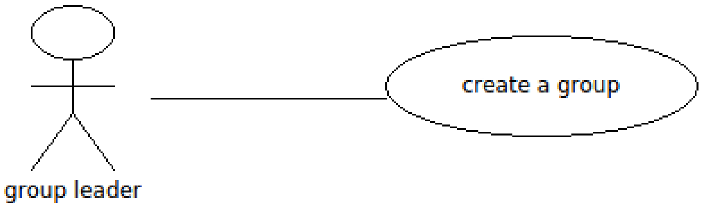
Initial step-by-step description:

1. The user will sign in using the required credentials.
2. After authentication, user enters into his/her own profile.

   Xref: Section 3.1.2,Sign in

2.3.3 Use Case: Create Group.

Diagram:



Brief description:

Before this use case can be initiated, the user has already registered on the system. The user will create a group and add friends.

Initial step-by-step description:

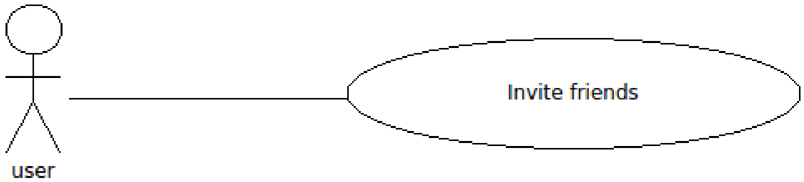
1. Any users who wish to use this application will register with his/her name and email id.
2. The system will display an option to create the group.
3. The member who creates the group will be considered as the group leader.
4. The group leader is also an user.

Xref: Section 3.1.3, Create group

2.3.4 Use case: Invite friends

This use case extends the 'Create group' use case.

Diagram:



Brief description:

To create friend list in IOU, an invitation will be sent to a person using their email id.

Initial step-by-step description:

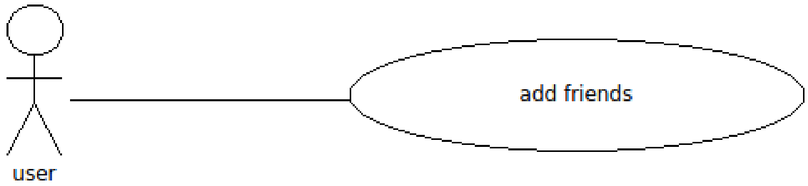
1. The IOU system provides the facility to make friends by displaying the option ‘Invite friend’s.
2. Apart from the group, a display called ‘friend list’ will be shown where the invited friends will be located.
3. Any member of IOU can add any person to the friend list by knowing his/her email-id.
4. After acceptance of the invite request, the names will be added into the database.

Xref: Section 3.1.4, Invite friend’s.

2.3.5 Use case: Add Friends

This use case extends the Create group use case.

Diagram:



Brief description:

The user will add friends in the group after creating it.

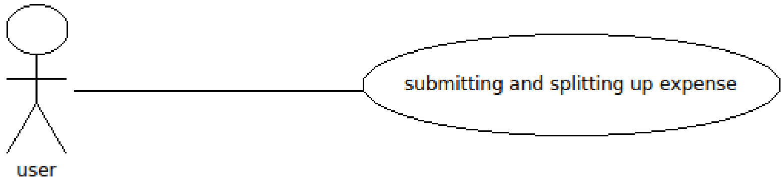
Initial step-by-step description:

1. After creating the group, the system gives the option to add friends into the group.
2. Any member of the group can add friends from the friend list.
3. The invited friend (in friend list) will be included into the group.
4. Group name and members will be displayed
5. Friends can be added to any group at any moment by the members of the group.

Xref: Section 3.1.5, Add friends

2.3.6 Use case: Submitting an expense.

Diagram:



Brief Description:

The members will submit the expenses of a tour in the IOU application.

Initial step-by-step description:

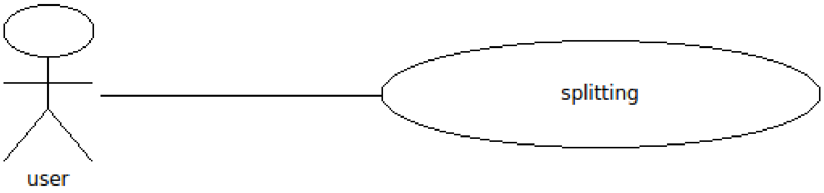
1. The user will make an entry of the total amount spent in a particular trip.
2. The system will show an option to name the expense.

3. The trip name and the group members will be displayed.
4. The group can be edited anytime.

   Xref: Section 3.1.6,Submitting and splitting up the expense

2.3.7 Splitting up the expense.

Diagram:



Brief description:

The system divides the amount among the group members with request to the input.

Initial step-by-step description:

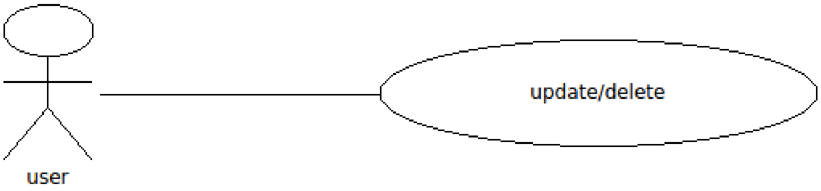
1. The system will display an option to split up the total expense.
2. The options may include the total amount and whether to divide equally or among specific persons.
3. The user will submit an option.
4. The system will generate the amount to be paid accordingly.
5. The amount will be added to the members account.
6. The members can check their balance.

   Xref: Section 3.1.6,Submitting and splitting up the expense

2.3.8 Use Case: Update/delete

This use case extends the submitting and splitting expenses use case.

Diagram:



Brief description:

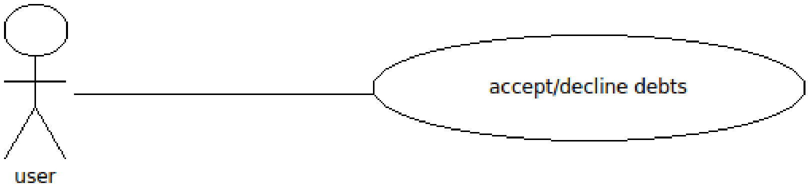
The user can update or delete the expenses added by him/her.

Initial step-by-step description:

1. After submitting the expense, the system provides the option to modify or delete it.
2. Only the person who will submit the expenses can modify/delete it.
3. The changes will be visible to the whole group.

   Xref: Section 3.1.7,Update/delete

2.3.9 Use case: Accept/decline the debt.



Brief description:

After the debt has been added into the account, the members may accept or decline the payment they owe to others.

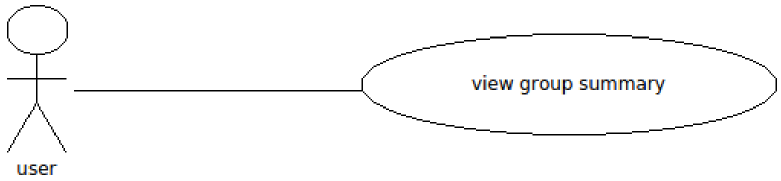
Initial step-by-step description:

1. The members can view the account details i.e the amount they owe and owed.
2. The system divides the payment and notifies the person.
3. The member can accept or decline the debt.
4. After acceptance of a debt, the amount will be deducted from the person who owes and will added to the person who owed.
   Xref: Section 3.1.8,accept/decline the debt

2.3.10 Use case: View Group summary

This use case extends the view summary use case.

Diagram:



Brief Description:

This option displays the total group summary.

Initial step-by-step description:

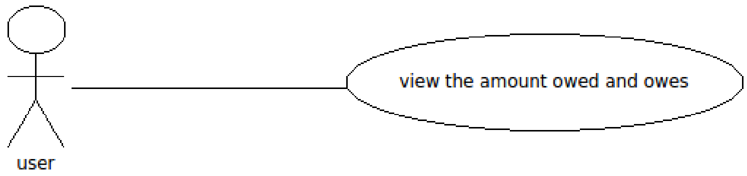
1. The system will display an option to view the transaction(off cash) chart.
2. Whenever any member requires viewing the complete history for the certain period, it can be viewed by clicking transaction chart option.
3. This option will provide a calendar in order to select the 'from and to' date for the specific period.
4. The total debts, acceptance and denial among each person within the group will be shown.

Xref: Section 3.1.9, Group summary

2.3.11 Use case: view the amount owed/owes.

This use case extends the view summary use case.

Diagram:



Brief description:

The members of the group can view the amount owed from others and the amount they owe to others.

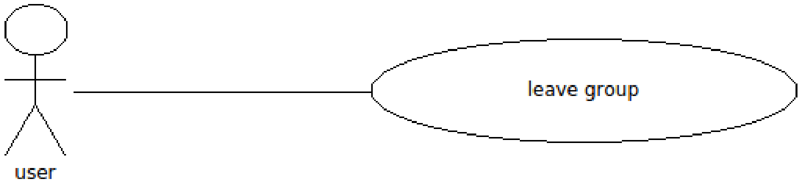
Initial step-by-step description:

1. After splitting up, the amount to be paid by each other will be added to their profile.
2. Every member can view the amount he/she owes to others and the amount owed from others.

   Xref: Section 3.1.10,Account summary

2.2.12 Use case: Leave group

Diagram:



Brief description:

The user can leave any group after getting acknowledgement from the group leader.

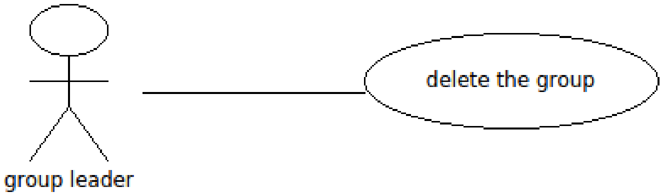
Initial step-by-step description:

1. Any member of a group can leave the group whenever he/she wants to after being acknowledged by the group leader (i.e) if the group leader gets any request from a member to remove him/her, then he/she can approve the request after checking their profile.

   Xref: Section 3.1.11, Leave group

2.3.13 Use case: Deleting the group.

Diagram:



Brief description:

The group leader will be having the rights to delete the group.

Initial step-by-step description:

1. The group leader can delete the group.
2. If he/she wishes to do so, a notification must be sent to all the members of the group.
3. After getting acknowledgement from all the members, the leader can delete the group.
4. The deleted history will be saved for future reference.
5. In case if any member wishes to leave the group, it is possible but deleting the entire group can be done only by the group-leader.

   Xref: Section 3.1.12, Delete group

2.4 User Characteristics:

It is assumed that the user is familiar enough with the computer to operate the browser, keyboard, mouse and is capable of browsing to, from and within simple websites.

2.5 Assumptions and Dependencies:

The protocol will be made to design the product which focuses on the features which are most important to the users.

3.0 Specific Requirements:

3.0.1 External Interfaces:

No external Interfaces will be required for this product.

3.0.2 Performance Requirements:

Performance requirements define acceptable response times for system functionality.

- The load time for the user interface screens shall take no longer than two seconds.
- The log in information shall be verified within five seconds.
- Queries shall return results within ten seconds.

3.1 Functional Requirements:

3.1.1 Sign up

Use Case Name	Sign up
XRef	Section 2.3.1 Use case: Sign up
Trigger	The new user requests to sign him up in the application.
Precondition	The person must have internet knowledge.
Basic Path	1. The user must specify unique name and email id. 2. All the required information should be satisfied by the new user.
Alternative Paths	Data validation is mandatory. If any required field is blank, the system will instruct the user to fill it up.
Post condition	The authenticated user will be added to the database.
Exception Paths	The attempt may be abandoned any time.
Other	The user’s name will be registered into the database for the recurrent use of the application.

3.1.2 Sign in

Use Case Name	Sign In
XRef	Section 2.3.2 Use case: Sign in
Trigger	The user wants to make use of the application.
Precondition	The user should be a registered user.
Basic Path	1. The user opens the site and logs in. 2. The user enters the application after authentication.
Alternative Paths	No alternate path for sign in.

Postcondition	The application will be available for the user.
Exception Paths	The attempt may be abandoned any time or if the authentication fails.
Other	The user can log into the system only after registering in the system.

3.1.3 Use case: Creating a group.

Use Case Name	Group Creation
Xref	Section 2.3.3 Use case: Create group
Trigger	When the User hits create group button.
Precondition	User have to log into the IOU system.
Basic Path	1. The user chooses add button from the group menu. 2. Group name is required. 3. The group will be created by hitting the create group button.
Alternative Paths	In step 2 if user leaves the space blank, then the user will be instructed to add a new name.
Postcondition	The group will be created. Members can be added now.
Exception Paths	The attempt may be abandoned anytime.
Other	While creating the group, user will be asked to add the members also. The member may be added later too.

3.1.4 Invite Friend's :

Use Case Name	Invite friends.
XRef	Section 2.3.4 Use case: Invite friends
Trigger	The user requests the application to invite friends into IOU.
Precondition	The user must have a registered account in IOU.
Basic Path	1. The user opens the IOU account. 2. Inviting a person using his/her email id. 3. After the acceptance of invite, the member will be added

	into the friend list. 4. The friend list will be displayed. 5. The members from the friend list can later be added into a group.
Alternative Paths	No alternate path for Inviting a member.
Post condition	The application will successfully display that person's name in the friend list.
Exception Paths	The attempt may be abandoned any time or if the intended person is not present in that user's friend list.
Other	N/A

3.1.5. Add friends

Use Case Name	Add friends.
XRef	Section 2.3.5 Use case: Add friends
Trigger	The user requests the application to add friends into the group.
Precondition	The user must have a registered account in IOU.
Basic Path	6. The user opens the group menu. 7. Select the group. 8. Selects the option to add a member. 9. The friend list will be displayed. 10. The member will be added to the group from the friend list.
Alternative Paths	No alternate path for adding a member in a group.
Post condition	The application will successfully add that person to that group.
Exception Paths	The attempt may be abandoned any time or if the intended person is not present in that user's friend list.

Other	If the requested person is not the user’s friend list then the user can invite the person by applying the use case 3.1.4.

3.1.6 Submitting and splitting up the expense

Use Case Name	Submitting and splitting up the expense
XRef	Section 2.3.6 Use case: Submitting an expense Section 2.3.7 Use case: Splitting up the expense
Trigger	The user submits the amount spent.
Precondition	The user should be a registered user.
Basic Path	1. The user submits the amount spent. 2. The system provides option to split the total expenses 3. Options will include equally, among specific persons etc. 4. After selecting an option, the system generates the individual amount to be paid.
Alternative Paths	None.
Postcondition	The generated amount will be displayed in each member’s profile.
Exception Paths	The attempt may be abandoned any time.
Other	N/A

3.1.7 Update/delete

Use Case Name	Update/Delete expenses
XRef	Section 2.3.8 Use case: Update/delete
Trigger	When a user wants to update or delete a previously claimed expense.

Precondition	The user must be a claimant and the claim must be raised before the updation/deletion.
Basic Path	1. Enter in to the group in which the user raise his/her claim. 2. Select the previously done transaction. 3. Then can edit the amount and other detail.
Alternative Paths	1. After step 2 in basic path user can delete that claim also.
Post condition	Whenever the user update or delete the transaction the funds will be debited or credited to the other member accordingly.
Exception Paths	The attempt may be abandoned at any time.
Other	N/A

3.1.8 Accept/decline

Use Case Name	Accept/Decline debts
XRef	Section 2.3.9 Use case: Accept/decline the debt
Trigger	Any of his group member claims a debt on him.
Precondition	The claimant and the user should be members of the same group.
Basic Path	1. When the claimant will claim against the user then a notification will come in the user's account. 2. After viewing the claim, user can accept the claim and settle up the claim (the amount will be added to his/her account).
Alternative Paths	1. After step 1 in basic path user can decline that claim also.
Post condition	If the user accepts the claim then the amount will get added into the claimant account. Otherwise a rejection message will be sent to the claimant.
Exception Paths	The attempt may be abandoned at any time.

Other	N/A

3.1.9 Group summary

Use Case Name	View full transaction chart
XRef	Section 2.3.10 Use case: Group summary
Trigger	The user selects view full detail option.
Precondition	The user should be a member of the group.
Basic Path	1. The user should go to the group menu. 2. Select a particular group. 3. Select view full details option.
Alternative Paths	1. Select the user's home. 2. Select account trends. 3. View full detail.
Post condition	The detail of the transactions of group members will be viewed as summarized data for the mentioned period.
Exception Paths	The attempt may be abandoned at any time.
Other	By following the alternating path. The user can see only the transactions where he/she is related.

3.1.10 Account summary

Use Case Name	Amount Owed and owes
XRef	Section 2.3.11 Use case: View the amount owed/owes
Trigger	The user selects all expenses from his home.
Precondition	The user should log in to his/her profile.

Basic Path	1. The user log in to the system. 2. Go to the home or dashboard. 3. The total amount he owes to others and the amount he owed from others will be displayed.
Alternative Paths	1. After option 2. 2. Select all expenses option.
Post condition	Individual account summary will be shown.
Exception Paths	The attempt may be abandoned any time.
Other	Basic path will give a grid view with total amount owed and owing amount. And alternating path will give a list view.

3.1.11 Leave group

Use Case Name	Leave group
XRef	Section 2.3.12 Use case: Leave group
Trigger	A member wishes to leave the group
Precondition	The user should be a registered user.
Basic Path	1. Any member may wish to leave the group anytime. 2. Leave group option will be displayed. 3. After choosing the leave group option, a notification will be sent to the group leader. 4. In case, if the person requested doesn’t have any unsettled debts, then the leader will acknowledge his/her request. 5. After getting approval, the member

	will be removed from the group.
Alternative Paths	None.
Post condition	The history till date of leaving can be viewed.
Exception Paths	The attempt may be abandoned any time.
Other	N/A

3.1.12 Delete Group

Use Case Name	Delete the group
XRef	Section 2.3.12 Use case: Deleting the group
Trigger	The user selects the delete group option.
Precondition	The user should be the leader of the group. Prior notification should be sent to all the members of the group before deleting.
Basic Path	1. The user opens the group. 2. The settings options will be displayed on the screen. 3. The user selects the delete group from the settings menu. 4. Notification will be sent to all the members of the group. 5. Group will get deleted after getting approval from all the members.
Alternative Paths	None.
Postcondition	The group leader can't delete the group unless he/she gets approval from the other members.
Exception Paths	Deletion of a group without confirmation.
Other	This use can be edited. For the confirmation part. If there is a better way to do it, then that thing will be followed.

3.2 Logical database requirements:

All data will be saved in the database: user accounts and profiles, messages etc. The database allows concurrent access and will be kept consistent at all times, requiring a good database design.

3.3 Design Constraints:

The system should work on most home desktop and laptop computers which support javascript and html. The system will be intended to run on firefox 4 and above, google chrome, Internet Explorer 8 and above.

3.4 Software System Attributes:

3.4.1 Reliability:

We mean that the system produce correct result degree of accuracy, in this case the system has integrity, the system must be something that can be depend on, so that the user can choose the wanted data and the system must respond the expected result and if another user do the same step, the system must generate the same expected and desired result.

3.4.2 Availability:

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved with the MySQL server and saved by the administrator.

3.4.3 Security:

The system will have the factors which protect it from accidental or malicious, since eac h user has special user name and password, they can keep track of individual data.

3.4.4 Maintainability:

The system is able to be maintained in future, as any new progress or development take place in the system.

3.4.5 Portability: The system can run on more than one machine, and generate the correct result in all cases.