Software Requirements Specification (SRS) Document

<iOS App development of SuperGenie, SSAD_team48,
Shaleen Garg, Ashutosh Ranjan, Shreyash Shriyam>

Brief problem statement

SuperGenie is an on demand personal assistant for ecommerce. Buying products or services online is a pain. User has to make a lot of decisions and choose between different websites to get the best deal. SuperGenie solves that problem by making it as simple as chatting to order what you want.

We are building a hybrid iOS app that will allow users to place orders via chat. It will have the following components:

Basic Features:

1. Building a realtime chat client.
2. Proper bot response integration in chat.
3. Building multiple chat rooms/heads.
4. Display of options regarding the request.
5. Payment gateway

System requirements

1. Angularjs- Structural framework for dynamic web pages
2. Bootstrap- HTML, CSS, Javascript framework for developing responsive mobile-first websites.
3. Javascript- A scripting language to enable web authors to design interactive websites

Users profile

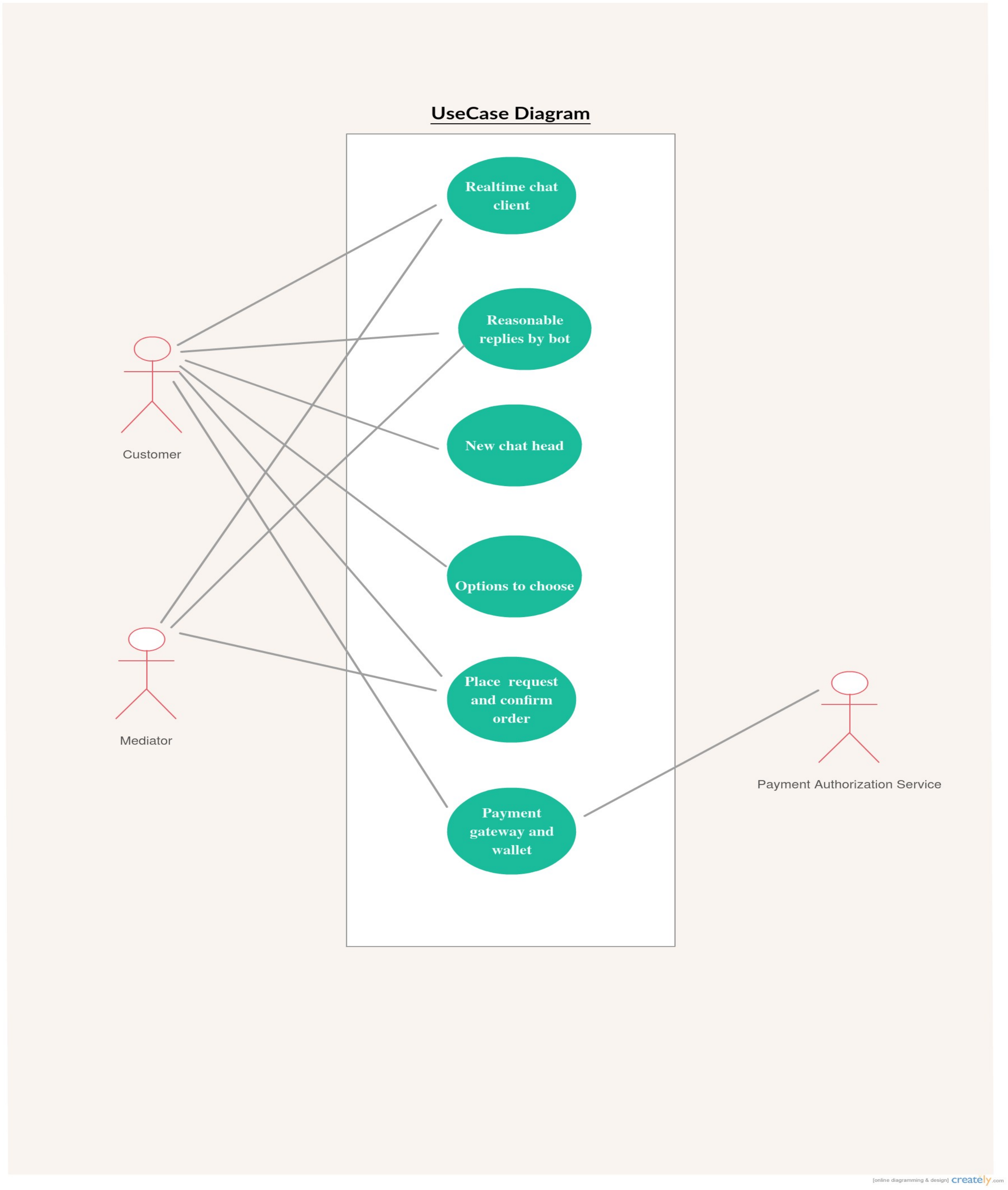
This application can be used by people of all ages and genders who can read english and have basic knowledge of operating a smartphone. This application is very useful for people who want to save the trouble of getting into long queues to get movie tickets, railway tickets, flight tickets, groceries etc without even worrying about the payment problems.

Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	Building a realtime chat client	We want to build a chat client similar to gmail chat or facebook chat but with lesser features needed only for our SuperGenie project. The customer should be able to send a message which would be received by the mediator present who would do whatever is needed.	1
2.	Proper bot response integration in chat.	The AI bot used in the chat should be able to give reasonable answers to the questions asked and the response should be prompt.	1
3.	Building multiple chat rooms/heads.	There should be more than one chat head.	2

4.	Display of options regarding the request.	The bot should be able to all the options regarding the request made by the user.	2
5.	Payment gateway	The options should have a payment option attached to them from where we can be redirected to a place where we can mak the payment.	

Use case diagram


UseCase Diagram
Realtime chat client
Reasonable replies by bot
New chat head
Options to choose
Place request and confirm order
Payment gateway and wallet
Customer
Mediator
Payment Authorization Service
[online diagramming & design] creately.com

Use case description

Use Case Number:	1
Use Case Name:	Realtime chat client
Overview:	We want to build a chat client similar to gmail chat or facebook chat but with lesser features needed only for our SuperGenie project. The customer should be able to send a message which would be received by the mediator present who would do whatever is needed.
Actors:	Customer Mediator
Pre condition:	Internet network should be available to the customer.
Flow:	Main (success) Flow: 1. The customer types the message and sends it in the chat client. 2. Mediator on the other side receives it.
	Alternate Flows: No alternate flow..
Post Condition:	The mediator should now be able to give the options to the customer regarding the request.

Use Case Number:	2
Use Case Name:	Proper bot response integration in chat.
Overview:	The AI bot used in the chat should be able to give reasonable answers to the questions asked and the response should be prompt.
Actors:	Customer Mediator
Pre condition:	The customer has typed in and submitted some query.
Flow:	Main (success) Flow: 1. The customer puts in some query. 2. The bot replies to that query promptly. 3. The customer is satisfied and moves forward with the conversation.
	Alternate Flows: 1. The customer types something that can't be understood by the bot. Then the bot asks the same question back again.
Post Condition:	The customer has made the request and bot is able to search for options.

Use Case Number:	3
Use Case Name:	Building multiple chat rooms/heads.

Overview:	There should be more than one chat head.
Actors:	Customer
Pre condition:	The customer wants to start a new chat.
Flow:	Main (success) Flow: 1. The customer wants to start a new chat. 2. The customer can click on the + button to make a new chat. 3. Or the customer can go back to the chat heads already made and start back again.
	Alternate Flows: No alternate flow.
Post Condition:	The customer is able to successfully make new chat heads.

Use Case Number:	4
Use Case Name:	Display of options to choose from
Overview:	The bot should be able to all the options regarding the request made by the user.
Actors:	Bot

	Database
Pre condition:	The request has been made by the customer and the database contains options regarding the request.
Flow:	Main (success) Flow: 1. The customer makes the request. 2. The bot searches the database. 3. The database returns the options.
	Alternate Flows: The request is not available in the database. Then the bot returns “no results found”.
Post Condition:	All the options are displayed from which the customer can choose from.

Use Case Number:	5
Use Case Name:	Enter the name of Use Case
Overview:	The options should have a payment option attached to them from where we can be redirected to a place where we can mak the payment.
Actors:	Payment authorization services Customer

Pre condition:	The customer chooses one of the options put forward by the bot.
Flow:	Main (success) Flow: 1. The customer clicks on the option of his choice. 2.The customer make the payment using the payment authorization service.
	Alternate Flows: No alternate flow
Post Condition:	The transaction is successful.