



SOEN 6481 - Software System Requirements Specification

Fall 2019

Requirements Analysis and Elicitation for Ticket Vending Machine

Project Report

Deliverable 2

Presented to

Instructor: PANKAJ KAMTHAN

By
Team G

Nirav Ashvinkumar Patel - 40081268
Rohan Deepak Paspallu - 40093648
Jingya Pan - 40044079
Divya Pandit - 40087471
Koshaben Patel - 40094385

Github URL : <https://github.com/niravjdn/SRS-Project>

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
2	User Stories	4
2.1	Global Constraints	4
2.2	iGo User Stories	4
2.2.1	iGo Registration	4
2.2.1.1	Negative user story of iGo registration	5
2.2.2	iGo Login	6
2.2.2.1	Negative user story of iGo login	6
2.2.3	Reset Password	8
2.2.4	Link Opus Card with iGO	9
2.2.5	Unlink the OPUS card from iGo account	11
2.2.6	View OPUS card balance in iGo	12
2.2.7	Load OPUS Card via iGo	12
2.2.7.1	Negative user story of iGo load OPUS card	12
3	Persona	14
4	Traceability Matrix	20
5	Implementation	22
5.1	Introduction	22
5.2	Quality Attributes	22
5.3	User Story 1,2 : Registration using email	23
5.4	User Story 3,4 : Login	26
5.5	User Story 5,6: Reset Password	28
5.6	User story 7,8 : Link and Unlink Opus Card	30
5.7	User story 9: View OPUS Card Balance	33
5.8	User story 10: Load OPUS Card	33
6	Team Contribution	35

Chapter 1

Introduction

This document provides detailed description of **iGo**, an electronic ticketing kiosk of public transportation which shall be designed to facilitate individuals of Quebec, Canada. Description of iGo is realised by including user stories, personas, traceability matrix.

1.1 Purpose

The purpose of this document is to provide traceable **user stories** for an online Ticket Vending Machine system (iGo) in Montreal, Quebec, Canada. Elicitation of persona is relevant to user stories. The data gathered from conducting interviews serve the basis of the user stories in this document. User stories prioritisation is realised by collecting information from external stakeholders which they would like to see in iGo application. iGo system is implemented by keeping quality concerns in mind. This document also provides glossary (**GIGO**) that supports all artifacts of iGo. This artifacts in this document is informed by each other, as well as by artifacts of deliverable 1.

1.2 Scope

This document applies to the Online Ticket Vending Machine (called iGo) in Montreal, Quebec, Canada, mainly for metros and buses. iGo plays as an online platform allowing **travellers** to top up their **OPUS** cards and manage **STM** transactions themselves. iGo does not include the development and maintenance part of STM system and physical STM Ticket Vending Machines in STM offices. Scope of implementation is limited to iGo positive user stories.

Chapter 2

User Stories

Use case model helps in understanding behaviour resulting from possible interactions with the system. However, User story represents functionality of a system[3]. As functionality of a system is exposed, user story can help during implementation part. Positive and negative user stories are included. However, negative user story is not designed or implemented and has no acceptance criteria[4].

2.1 Global Constraints

1. iGo system is developed to connect with STM system via [REST APIs](#). Therefore, the global constraint of all user stories is the fact that it is assuming the core business logic is handled properly via STM system. iGo will request and receive response from STM with its precise workflow, however, the handling component from STM system is not managed and developed, tested by iGo team.
2. End user is expected to test and use iGo system by modern web browsers such as Google Chrome, Microsoft Edge, Mozilla Firefox, Safari because iGo is developed by Single Web Application in [Spring Framework](#) using [Java](#). It is supposed to work efficiently with modern web browsers.
3. iGo system is not accessible if user does not have internet connection, therefore it is expected by iGo team that user has internet connection if they want to use services from STM system or iGo sysrem.

2.2 iGo User Stories

The following are the user stories[[userstory](#)] of iGo [USIGO](#) which includes both positive and negative stories.

2.2.1 iGo Registration

User Story 1: As an unregistered user of iGo system, I wish to create a new account on iGo website by providing details such as name, valid email id, password and receiving an email that contains link to verify identity in secure manner. Upon completing registration process I intend to log into the

system by providing credentials and use services offered by iGo.

2.2.1.1 Negative user story of iGo registration

Malicious actor does flood attack on iGo system to make iGo unable to respond to legitimate users.

Priority: High

Acceptance Tests

- *Email Format Rule: The rule is an email string must contain the symbol “@” with characters and ”@” doesn’t appear at the end of a string.
- *Password Format Rule: User’s password must be more than 8 characters.

ID	Given	When	Then
1.1	User’s email id = “nirav@gmail.com”, User’s valid password = “nirav@6481”	User enters inputs of Registration Form in IGo website to create an account.	iGo validates whether input data of fields email id, password are in the correct format. A new iGo account is created.
1.2	User’s email id = “kosha”, User’s valid password = “kosha@6481”	User enters inputs of Registration Form in iGo website to create an account.	iGo validates data of input field email id is invalid. iGo displays an error message on iGo register form. Creation of new iGo account is denied by iGo system.
1.3	User’s email = “rohan@gmail.com”, User’s password = “1234”	User enters inputs of Registration Form in iGo website to create an account.	iGo validates data of input field password is less than 8 characters. iGo displays an error message on iGo register form. Creation of new iGo account is denied by iGo system.

User Story 2:

As an existing iGo user, I try to create a new account with my previous registered email id so that I can not create a new iGo account and appropriate error message is displayed stating relevant information.

Priority: High

Acceptance Tests

ID	Given	When	Then
2.1	Email = “divya@gmail.com”, Password = “password1234”	User provide inputs of Registration Form in iGo website to create an account.	iGo system verify whether user’s email id already exists in iGo database which return a failed status message stating entered email id is associated with existing iGo account. iGo displays on iGo sign up form, stating that “This email exists in our system. Please login.”. Creation of new iGo account is denied by iGo system.

2.2.2 iGo Login

User Story 3: As an existing iGo user, I wish to login iGo website by providing credentials to use services offered by iGo.

2.2.2.1 Negative user story of iGo login

Malicious actor attempts to steal credentials of legitimate users to get access of iGo account.

Priority: High

Acceptance Tests

- *Email Format Rule: The rule is an email string must contain the symbol “@” with characters and ”@” doesn’t appear at the end of a string.
- *Password Format Rule: User’s password must be more than 8 characters.

ID	Given	When	Then
3.1	Email = “jingya@gmail.com”, Password = “soen@6481” ”	User provide inputs of Login Form on iGo website.	iGo validates whether entered email id and password adhere to email format rule, password format rule. iGo verifies login credentials using iGo database. iGo logs user in successfully and saves the login session of a user on iGo system. iGo displays login successful message and navigates to user dashboard.
3.2	Email = “kay1045”, Password = “kayy@1045”	User provide inputs of Login Form on iGo website.	iGo validates that entered email id doesn't comply with email format rule. iGo displays a validation error message at Login Form on iGo website.
3.3	Email = “nirav@gmail.com”, Password = “123”	User provide inputs of Login Form on iGo website.	iGo validates that password is more than 8 characters. iGo displays a validation error message at Login Form on iGo website.

User Story 4: As an existing iGo user, I want to login with wrong credentials so that I will receive an error message and iGo system will not log me in.

Priority: Medium

Acceptance Tests

- *Email Format Rule: The rule is an email string must contain the symbol “@” with characters appending to the symbol “@”.

ID	Given	When	Then
4.1	Email = “user@gmail.com” Password = “wrongpassword”	User fills in Login Form on iGo website.	iGo validates that email and password are in the correct format. iGo verifies login credentials in database returns message that credentials are wrong. iGo displays a wrong message to a user on iGo login form.
4.2	Email = “wrongEmail@gmail.com” Password = “wrongpassword”	User fills in Login Form on iGo Website.	iGo validates that email and password are in the correct format. iGo verifies login credentials in database returns message that credentials are wrong. iGo displays a wrong message to a user on iGo login form.

2.2.3 Reset Password

User Story 5: As an existing iGo user, I want to reset my password so that I can get back in to iGo site if I forget my password.

Priority: High

Acceptance Tests

ID	Given	When	Then
5.1	User is registered with valid email address.	User clicks Forget password link under Login Form. iGo displays Forget password form where user enters registered email address and press Submit.	iGo validates an email address is in the correct format. iGo verifies whether email is registered in iGo database. iGo sends an email with reset password link to user's registered email. The reset password link is valid for 24 hours.

5.2	User is not registered on iGo.	User clicks Forget Password link under Login Form. iGo displays Forget Password form where user enters registered email address and press Submit.	iGo validates an email address is in the correct format. iGo verifies with iGo database if Email is registered. STM system returns with status stating Email does not exist. iGo displays a message that “An email does not exist” to a user at Forget password form on iGo website.
5.3	User enters invalid email on Forget Password Form.	User clicks Forget password link under Login Form. iGo displays Forget Password form. User enters an invalid email.	iGo validates an email is not in the correct format. iGo displays a message that “An email does not exist” to a user at Forget password form on iGo website.

User Story 6: As an existing iGo user, I want to be facilitated to change the password in a secure way through email id sent to my inbox in case if I forget a password or want to create a strong password.

ID	Given	When	Then
6.1	User receives an email with reset password link.	User clicks Reset Password link within 24 hours receiving an email.	iGo displays a Change Password form on iGo website where user enters new password. iGo changes user's new password and updates iGo database.
6.2	User receives an email with reset password.	User clicks Reset Password link after 24 hours receiving an email.	iGo displays an error message on iGo website.

2.2.4 Link Opus Card with iGO

User Story 7: As an existing iGo user having physical OPUS card, I want to link my OPUS card to my iGo account after I log into the iGo system, so that I can reload my OPUS card anytime and anywhere with iGo in order to save time going to nearest metro station to reload the OPUS card (Time consuming and tedious during emergency or peak hours).

Priority: High

Acceptance Tests

- *Rule 1: One OPUS card can be linked to one iGo account. However, the iGo account may need Valid Opus card number in order to link it with the account.
- *Rule 2: Valid Opus Card number consists only of numbers(digits). Thus the fields to enter Opus Card number in iGo system wont accept input with special characters or letters.

ID	Given	When	Then
7.1	User enters valid OPUS card number printed on physical OPUS card. Example: OPUS Card Number = "1234567890123"	Once user is logged in to iGo system, User can now view dashboard. User wish to link opus card through navigation bar. User now needs to fill a Link Opus Form in order to link his opus card.	After entering valid Opus card number, iGo verify Opus card number in iGo database. iGo will now validate OPUS card number to check if its the required format.(*Rule 2). iGo now verifies with STM system if OPUS card provided is valid or not.(*Rule 1). After successful linking of Opus Card with iGo system, it will display a successful message stating "Opus card is linked successfully with iGo system" to a user.
7.2	If User enters Opus card number already linked to the same/another active iGo account.	User enters opus card number already associated to same/another iGo account and click on submit button.	iGo will validate if OPUS card number is in the valid format or not.(*Rule 2). After successful validation iGo will verify with STM system to check whether entered Opus card number is already linked with any other active account or not. If found linked with same/another account, iGo system returns message stating "OPUS card is already linked to another active iGo account. Please do link another valid Opus card. Sorry for the inconvenience caused".

7.3	If User enters Invalid Opus card. Example: OPUS Card Number = "123123"	User clicks Link Opus Card and fills the form to link the card.	Once the form is submitted iGo validates that OPUS Card Number to check if its in valid format or not. (* Rule 2) iGo displays a message stating that "The OPUS card number entered to link is invalid. Please try again entering a valid one".
7.4	User enters invalid OPUS Card Number that is not issued by STM. Example: OPUS Card Number = "1234567891111232".	User clicks Link OPUS Card to display OPUS Card Form and submits the form. The entered Opus card number is invalid as it exceeds the length of valid opus card number.	iGo displays a message stating user that "The OPUS card number entered to link is invalid. Please try again entering a valid one".

2.2.5 Unlink the OPUS card from iGo account

User Story 8: As an existing iGo user, I wish to remove a linked OPUS card from my iGo account, to delete my information from iGo system. This can happen given circumstances user has moved or does not wish to use services from iGo system so he/she wants to delete the record from iGo database.

Priority: High

Acceptance Tests

ID	Given	When	Then
8.1	User is logged into iGo account and has a linked OPUS card that they want to unlink from that iGo account.	User wishes to manage his/her linked card to system and clicks on delete button on particular OPUS Card.	iGo displays a message saying "User has successfully unlinked his card from the iGo system account".

2.2.6 View OPUS card balance in iGo

User Story 9: As an existing iGo user, I can view linked OPUS card balance so that I can keep track of my usage.

Priority: Medium

Acceptance Tests

ID	Given	When	Then
9.1	User is logged into the iGo account. User has at least one linked OPUS card in his/her iGo account.	User clicks on the manage OPUS Card link and checks the balance icon .	The iGo system displays the OPUS card balance.

2.2.7 Load OPUS Card via iGo

User Story 10: As an existing iGo user having at least one linked OPUS card, I want to top up my OPUS card on iGo using visa/mastercard.

2.2.7.1 Negative user story of iGo load OPUS card

A fraudster wants to steal credit card information of legitimate users to cause them financial troubles.

Priority: High

Acceptance Tests

ID	Given	When	Then
10.1	User has at least one linked OPUS card number on iGo system. User has a valid VISA/Master card to pay money.	User clicks load button to recharge his opus balance which will direct him/her to web page where he/she can fill a Payment Form.	He/she can enter amount of money and VISA/Master card details. He/she can complete the payment, respective bank verifies the user entered values thus user receives prompted message "Payment is successful".

10.1.1	The user enters card details. The card number should be same as embedded on the card	When the user proceeds to make payment by entering the valid payment details.	Before successful payment the validity of the card and the payment information is checked and if the entered details are not valid the user is notified to enter the appropriate information.
10.1.2	The name on the card is given as input to the payment system.	When the user proceeds to make payment by entering the valid payment details.	Before successful payment the validity of the card and the payment information is checked and if the entered details are not valid the user is notified to enter the appropriate information.
10.1.3	The expiry date on the card is specified in MM/YY format.	When the user proceeds to make payment by entering the valid payment details.	Before successful payment the validity of the card and the payment information is checked and if the entered details are not valid the user is notified to enter the appropriate information.
10.1.4	The three digit CVV code is entered to make payment	When the user proceeds to make payment by entering the valid payment details.	Before successful payment the validity of the card and the payment information is checked and if the entered details are not valid the user is notified to enter the appropriate information.

Chapter 3

Persona

Persona can help establish an understanding of users, and serves a starting point for empathy towards users among software engineers[1].



Persona for user story 'iGo Registration'

Person Name: Vanessa Abrams

Job/Role Description: University student major in Painting and Drawing (BFA). She studies virtually approach to painting and drawing, from traditional oil painting to graphic novel production and 3D spatial installation.

Goals: Use the iGo system to simplify the opus card relevant operation, and save the time for recharge.

Abilities: Vanessa is a University student and familiar with the computer operation, and get access to the Internet almost everyday at home or campus.

Short narrative: Vanessa is university student Vanessa. She uses the computer almost everyday. In order to finish the school project, she needs to get some information from the website. She also uses computer to download the resources like the notes to review. Sometimes she likes to watch some movies in the weekend with her families. To recap, she is quite familiar with technology, and it is not hard for her to deal with the normal and regular operation with the computer.

In order to get access to the iGo system ,Vanessa visits the register page first and is notified to enter a valid email address and set a password which is defined to be more than 8 characters. For her, the operation is relatively easy and straightforward, she uses the email and set the password is a regular operation.



Persona for user story 'iGo Login'

Person Name: Daniel Humphrey

Job/Role Description: He is a model for certain brand, and he has worked in the field about 2 years after he graduated from the university.

Goals: Use the iGo system to simplify the card relevant operation, and save the time for recharge. His apartment is far from the working place, so the bus and the metro is the most common transportation means he uses.

Abilities: Daniel likes using the Instagram to upload his photo and communicate with his friends. He likes go out with his friends in the weekend and edit the photos he captured. The 'photoshop' or some other photo editing application is his most used software.

Short narrative: Daniel is also a frequent computer user, and he has the habit surfing on the internet, so the regular use and the photograph related software has no challenges for him.

In order to get the benefits of using the iGo system, Daniel need to get an account, and before using the service the system offered, he need to log in by entering the registered email address and the password. After the verification, Daniel is able to follow the instruction and use the system.

Persona for user story 'Link OPUS card with iGo'

Person Name: Charles Bass,

Job/Role Description: He is a Freelancer, and he is a big fan of various kind of music, sometimes he has live show in the pub with his band.

Goals: The show for the bar do not have a regular occasion. They will attend the activity if they receive the invitation from the pub or bar. In his daily life, the public transportation is the most frequently used way.

Abilities: Charles always use the social application to get contact with his friends. He likes listening to the music and watching movies online. Every week, he prefers to go out with his friend to the banlieue or some small towns not too far to find some inspiration. He likes writing lyrics and create the melody himself. He uses the software like editing the music and recording.

Short narrative: Charles likes surfing on the internet, So he is familiar with the operation and common process with the computer and get access to the internet.

In order to link the Opus card with the iGo system. Charles need to know the Opus card identification number and enter the valid email address. Definitely, the prerequisite of using the iGo system is the registration. Therefore, registering in the system is still needed.



Persona for user story 'Unlink OPUS card with iGo'

Person Name: Blair Waldorf,

Job/Role Description: She is an editor of the fashion design magazine. She has a good taste with cloths and interior designer.

Goals: She is an occasional user for the public transportation. Since she has a car, but when she needs to attend the party or the weather is bad, she prefers to take the bus or metro. With the help of the online system, she is able to simplify the transaction process.

Abilities: Blair likes to watch films acted by Audrey Hepburn. In the weekend, she prefers to watch the classic movie with her boyfriend. She also prefers shopping online, and find the outfits and the models that can be used in the magazine article. She also posted the photo through the 'instagram' and watch the 'fashionist' vlog to get to know the latest trend.

Short narrative: Blair likes surfing on the internet, because it is the way that she get to know the fashion trend. Therefore, the internet and the web is not unadaptable to her.

Different from the link operation. Unlink the Opus card with the iGo system is relatively easier. Blair need to log in the system before doing anything. iGo system will definitely make sure the user do the operation not accidentally and prevent user making mistakes.



Persona for user story 'View OPUS card balance in iGo'

Person Name: Darshan Patel.

Job/Role Description: He is a graduate student and developer in the animation production company for 6 months, he uses the .net framework in his daily work.

Goals: His work place is not far from his apartment, therefore he does not possess a private car. But in the weekend, sometimes he will go out with friends and go shopping to the Downtown area.

Abilities: Darshan always uses the social application to get in touch with his friends to talk about the game. He likes learning the tutorial online to learn the new animation technology. Every week, he prefers to go out with his friend to have a BBQ party. He likes self-learning and comes up with new ideas.

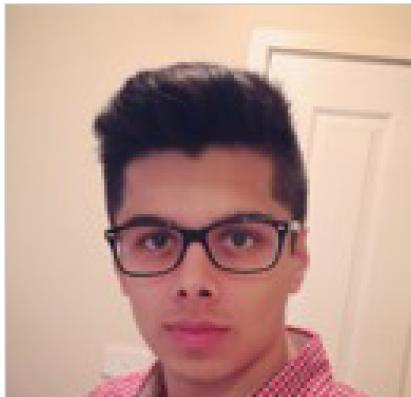
Short narrative: From Darshan's perspective, being a developer makes him really familiar with various computer online operation, even the advanced operation is not difficult for him. He can be generalized as expert user.

View the balance in the OPUS is a service provided by iGo system, Cyrus need to log in the system before doing the relevant operation. The function is really straightforward.

Persona for user story 'Reset password in iGo'

Patel Overleaf.png

PROJECT: CJM for Avia Travel



NAME

**Deep Patel, Graduate
Teaching Assistant**

PERSONALITY TYPE

Rational

Goals

I intend to provide solutions for unsolved problems. A wannabe Computer Scientist.

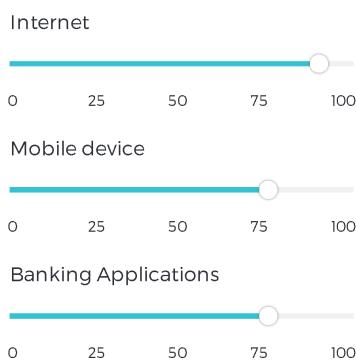
Demographic

♂ Male _____ 23 years
📍 Canada _____
Single _____
Graduate Teaching Assistant _____
Meticulous, Cautious, Friendly _____

Quote

“Every problem has a solution.”

Typical Usage



Motivations

- To browse information regarding services offered by STM
- To buy or recharge OPUS card

Background

23 y.o. Busy individual often working on problems. Typically spends time learning, surfing the internet. Heavily rely on technologies. Loves playing football, computer games. Likes to attend social events.

Needs

- 24/7 availability of services from organization
- Faster response
- Quick guidance
- Security while performing transactions online
- Assurance from organization when money is involved in any activity

Frustrations

- Time consuming activities
- Repetitive tasks
- Remembering passwords, username
- Animations
- Advertisements
- Excessive features
- Having to wait in long queue to recharge opus card

Persona for user story 'Load Opus Card in iGo'

Surve.png

PROJECT: CJM for Avia Travel



NAME

Aditya Surve, Graduate Student

PERSONALITY TYPE

Guardian

Goals

To bring the order in the world full of chaos.

Quote

If we clearly see, everything is either 0 or 1.

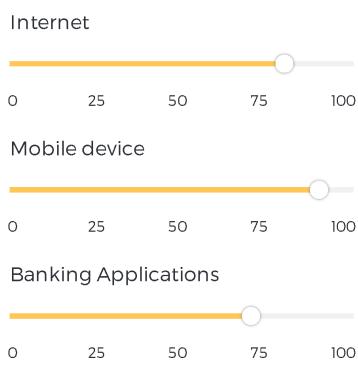
Demographic

♂ Male 24 years
📍 Canada
Single
Graduate Teaching Assistant
Logical, Orderly, Debater

Motivations

- To buy a ticket or order OPUS card
- To search the distance between source and destination

Typical Usage



Background

24 y.o. He studies software engineering. He likes to read about religions, technologies. Prone to by recently launched electronic gadgets. Suffers from social anxiety. He loves watching movies, hanging out with friends.

Needs

- Facility to order OPUS card from the internet
- Wants to recharge OPUS card performing online transactions
- Attractive User Interface
- Highly concerned with security
- Faster response while navigating through web application

Frustrations

- Having to wait for trivial tasks such as paying STM tickets
- Having to write credit card credential while performing online transactions
- Boring user interface
- Error pages displayed
- Not being able to find informations regarding questions online

Chapter 4

Traceability Matrix

The traceability matrix[2] of the iGo system **TMIGO**, is linking the requirements through the validation process. The traceability matrix is realized to ensure that all the requirements defined for the iGo system are tested and passed. The TMIGO is a table format containing the requirements and their description along with their source. TMIGO comes in handy when we need to trace the resource origin of the requirements in the long run.

In the process of creating the traceability matrix , we have ensured that the iGo system that we are creating is on the right track. It also helps us to determine if there are any extra unspecified functionality added to the requirements. Thus, we can manage the scope of the system and keep track of how the project is affected for every change in the development lifecycle.

ID	REQ ID	REQ TYPE	REQ DESC	REQ SRC	FLAG
1A	1	FN	Register with valid email creating email address	iGo system description in D1	Passed
2A	2	FN	Creating new account with previously registered email address	REQ ID 1	Passed
3A	3	FN	Login into the iGo system with registered email address	REQ ID 1	Passed
4A	4	FN	Invalid login	REQ ID 1	Passed
5A	5	FN	Reset password in case of wrong input credentials	User request in interview	Passed

6A	6	FN	Send reset password link via email to reset the password if the user forgets password	User request in interview	Passed
7A	7	FN	Link OPUS card to the iGo system	REQ ID 2	Passed
8A	8	FN	View OPUS card balance	REQ ID 3	Passed
9A	9	FN	Unlink the OPUS card from the iGo system	REQ ID 5	Passed
10A	10	FN	Reload OPUS card via Visa/Master card	REQ ID 4	Passed

Chapter 5

Implementation

5.1 Introduction

There are a set of user stories which are to be implemented by the project on iGo smart ticket vending machine. Here, the user stories that are to be implemented ensure that the quality attributes are attained by the system and the system implements all the user stories which can be used for improving the existing system. As mentioned previously, we assume that STM maintains REST API through which we can get various details regarding the customers and can perform functionalities on the database. The details contain valid information about the OPUS cards.

Here the technology which has been utilised by us for the purpose of the API is Java along with its tools. Moreover, for the development of User Interface, we have used the Spring Framework focuses on the MVC design pattern. The built web application is platform independent and can work on all the browsers which support HTML5 and CSS3. Lastly, the project is tested to work with various inputs and provide a desired output whenever there is an issue/error in the user input. All in all, it can be said that the system is well tested and ready to be deployed in the working environment and would fulfill all the requirements of the users.

Github Link for iGo system: <https://github.com/niravjdn/SRS-Project>

5.2 Quality Attributes

1. **Compatibility:** Our System shall provide co-existence as it will perform its required functions efficiently while sharing a common environment and resources with other systems, without detrimental impact on any other system and it will also be inter operable so that two or more systems can exchange and use information such as vehicle catalog information etc. This has been achieved by use of bootstrap library to render app on all the browser compatibly.
2. **Usability:** This system shall be easy to use for input preparation, operation, and interpretation of output and shall provide consistent user interface standards or conventions with our other frequently used systems. It will also be easy for new or infrequent users to learn to use the

system as It will provide learnability, operability, user error protection, accessibility and user interface aesthetics. Our system shall increase user confidence and satisfaction by providing user friendly navigation.

3. **Reliability** - iGo web app shall be available, accessible when required for use and operational as it is indented to operate despite the presence of hardware or software faults. Moreover, in the event of an interruption or failure, this system will recover the data directly affected and re-establish the desired state of the system. This can be achieved using persistent database.
4. **Security:** Our system shall provide confidentiality by making data accessible only to those authorized who have access through password privacy and prevent unauthorized access and modification of data. Thus, security can be characterized as a system providing non repudiation, confidentiality, integrity, assurance, availability, and auditing. This shall be achieved using spring security as a security layer. Along with authorization, we have also provided encryption facilities which can make the system more secure.
5. **Maintainability:** This system shall be maintainable because it is composed of discrete components such that a change to one component has minimal impact on other components. It shall also be reusable as module that is developed with high cohesion and low coupling can be used in more than one system, or building other assets. It will provide the effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified. Moreover, it will also be effectively and efficiently modified and tested without introducing defects or degrading existing product quality. This shall be achieved by using Spring MVC.
6. **Portability:** The system can be run on any tomcat server with minimum 128MB RAM. Hence, Considering this, System shall be portable.

5.3 User Story 1,2 : Registration using email

As a new iGo web user, I want to create a new account on iGo website by my valid email and a secure password. So that I can use that credentails to login later on iGo website with my registered credentials.

localhost:8080/registerUser

User Registration

Email Id

Password

User First Name

User Last Name

Address

Phone Number

REGISTER **RESET**

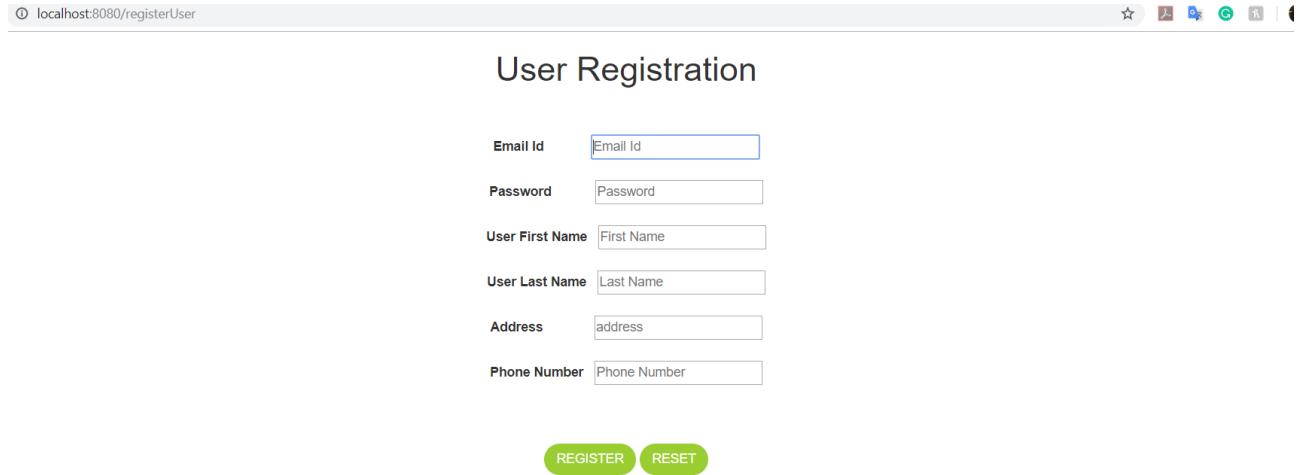
A screenshot of a web browser showing a user registration form. The URL in the address bar is 'localhost:8080/registerUser'. The page title is 'User Registration'. There are six input fields: 'Email Id' (containing 'Email Id'), 'Password' (containing 'Password'), 'User First Name' (containing 'First Name'), 'User Last Name' (containing 'Last Name'), 'Address' (containing 'address'), and 'Phone Number' (containing 'Phone Number'). Below the inputs are two green rounded rectangular buttons labeled 'REGISTER' and 'RESET'.

Figure 5.1: iGo web page to let new users to register into the system

localhost:8080/registerUser

User Registration

Email Id

! Please include an '@' in the email address. 'divya' is missing an '@'.

User First Name

User Last Name

Address

Phone Number

REGISTER **RESET**

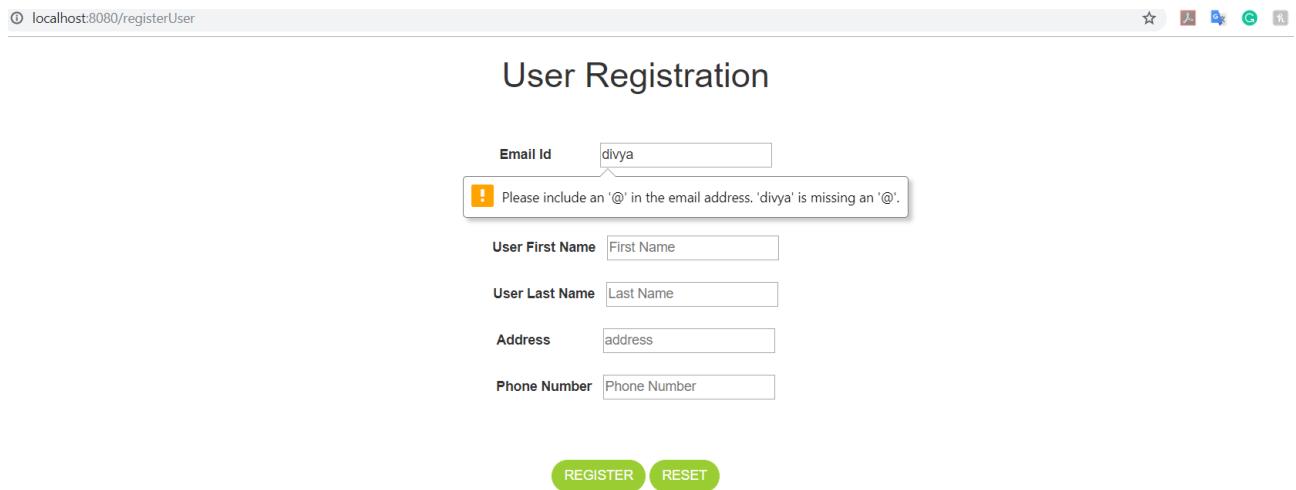
A screenshot of a web browser showing a user registration form with validation feedback. The URL in the address bar is 'localhost:8080/registerUser'. The page title is 'User Registration'. The 'Email Id' field contains 'divya'. A yellow warning box with a exclamation mark icon appears above the field, stating 'Please include an '@' in the email address. 'divya' is missing an '@''. Below the field are five empty input boxes for 'User First Name', 'User Last Name', 'Address', and 'Phone Number'. At the bottom are two green rounded rectangular buttons labeled 'REGISTER' and 'RESET'.

Figure 5.2: When user enters invalid email id format then the system will not let them create an account with iGo

localhost:8080/registerUser

User Registration

Email Id: divyapandit13@gmail.com

Password:

User First Name: ! Please match the requested format.
password should have minimum 8 characters

User Last Name: Last Name

Address: 936 Rue De l'Église, Apt 4

Phone Number: Phone Number

REGISTER **RESET**

This screenshot shows a user registration page with fields for Email Id, Password, User First Name, User Last Name, Address, and Phone Number. The Password field contains a short password. A validation message box appears over the User First Name field, stating 'Please match the requested format. password should have minimum 8 characters'. The other fields contain placeholder text or are empty.

Figure 5.3: When the user enters a password that contains less than 8 characters, the system does not let them create an account with iGo

localhost:8080/registerUser

User Registration

Email Id: divyapandit13@gmail.com

Password:

User First Name: ! Please fill out this field.

User Last Name:

Address: 936 Rue De l'Église, Apt 4

Phone Number: Phone Number

REGISTER **RESET**

This screenshot shows a user registration page with fields for Email Id, Password, User First Name, User Last Name, Address, and Phone Number. The User First Name field is empty. A validation message box appears over the User First Name field, stating 'Please fill out this field.'. The other fields contain placeholder text or are empty.

Figure 5.4: When the user needs to enter first name and last name before submitting details for new account in iGo

The screenshot shows a user registration form on a web browser. The URL is `localhost:8080/registerUser`. The form fields are as follows:

- Email Id: `divyapandit13@gmail.com`
- Password: `*****`
- User First Name: `Home`
- User Last Name: `sjas`
- Address: `936 Rue De l'Église, Apt 4`
- Phone Number: `51455`

A validation message box appears over the Phone Number field, stating: "Please match the requested format. Phone number must have 10 characters". Below the message are two buttons: `REGISTER` and `RESET`.

Figure 5.5: When the user enters a phone number that is not exactly matching 10 characters (Invalid Phone Number), the system does not let them create an account with iGo

The screenshot shows a user registration form on a web browser. The URL is `localhost:8080/userRegistered`. The message "This Email is already registered in the system!!" is displayed prominently in red at the top. The rest of the form fields are identical to Figure 5.5.

Figure 5.6: When the user tries to register using email used by already existing user.

5.4 User Story 3,4 : Login

An existing user want to login to iGo website with my registered email and password so that he/she shall be able to login to iGo website and use services offered by iGo.



Please Sign in

[Forgot Password?](#)

© 2019-2020

Figure 5.7: Login web page to let existing iGo user to enter valid credentials to login to the iGo system to perform various transactions



Please Sign in

Email or Password invalid, please verify

[Forgot Password?](#)

© 2019-2020

Figure 5.8: When the user enters invalid credentials that does not exist in the iGo database, then the iGo does not let the user login to their account



Figure 5.9: When the user enters valid credentials that exists in the iGo database, then the iGo redirects them to the dashboard of iGo account.

5.5 User Story 5,6: Reset Password

An Existing iGo user want to reset password. This might be possible if user has forgot the password or want to reset it.

A screenshot of a password reset form. At the top, there is a blue button labeled 'Go To Login Page'. Below it, the title 'Reset Password' is centered. A text input field is labeled 'Email' and contains a placeholder 'Email'. At the bottom of the form are two buttons: a grey 'Clear' button and a blue 'Reset Password' button.

Figure 5.10: Password reset page to let user enter valid email id

Go To Login Page

Reset Password

This email does not exist!

Email

Clear Reset Password

Figure 5.11: A page showing user that email does not exist if user enters invalid email which does not exist in iGo Database.

Go To Login Page

Reset Password

Request to reset password received. Check your inbox for the reset link.

Figure 5.12: A page showing that password reset link has been sent to email.



Figure 5.13: An email sent by system to user with password reset link

Enter new password:

Password

Clear Submit

Figure 5.14: A form to let user enter new password once he clicks on link received on the email

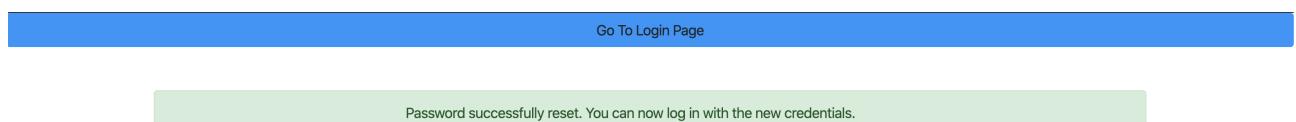


Figure 5.15: A page showing message that password reset has been successful and now user can login using new password.

5.6 User story 7,8 : Link and Unlink Opus Card

An existing iGo User want to link iGo Opus Card to his/her account. After login, the user can link and unlink opus card.

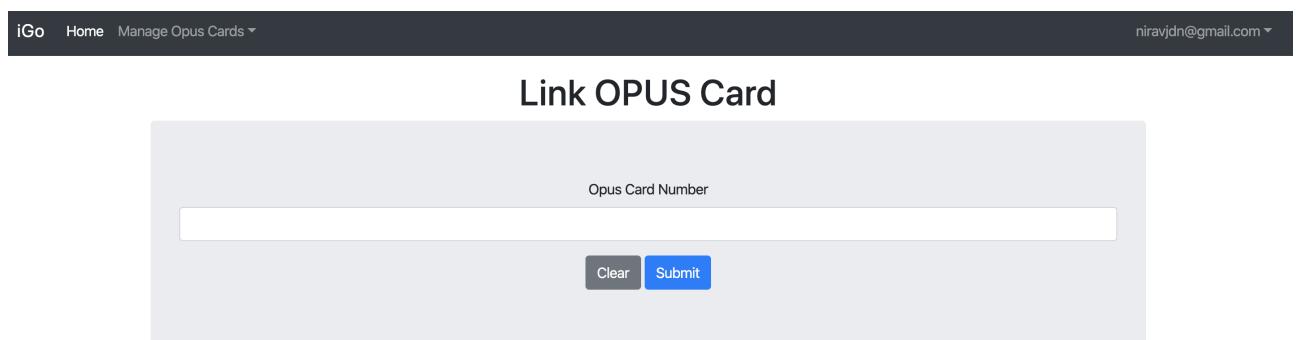


Figure 5.16: The web page to enter opus card details to link the card to the iGo account

The screenshot shows a web application interface for linking an OPUS card. At the top, there is a dark header bar with the text "iGo Home Manage Opus Cards" on the left and "niravjdn@gmail.com" on the right. Below the header, the main content area has a title "Link OPUS Card". A pink error message box contains the text "The OPUS card is already linked to another iGo account." Below the message is a form field labeled "Opus Card Number" with a placeholder text input box. Underneath the input box are two buttons: "Clear" (gray) and "Submit" (blue).

Figure 5.17: When the user enters opus card details that is already linked with another iGo account, the iGo system does not let them link that opus card in the account

The screenshot shows a web application interface for linking an OPUS card. At the top, there is a dark header bar with the text "iGo Home Manage Opus Cards" on the left and "niravjdn@gmail.com" on the right. Below the header, the main content area has a title "Link OPUS Card". A pink error message box contains the text "The OPUS card number is invalid." Below the message is a form field labeled "Opus Number" with a placeholder text input box. Underneath the input box are two buttons: "Clear" (gray) and "Submit" (blue).

Figure 5.18: When the user enter invalid Opus card details that is not in the STM database, the iGo does not let the opus card to be linked to the iGo system

The OPUS card has been linked successfully.

Opus Card Number

Clear Submit

Figure 5.19: When the user enters valid opus card details which is valid and not linked to any other iGo account, then iGo system lets the opus card to be linked to this iGo account.

Id	Number	Balance	Operation
2	1212121212126	10	

Figure 5.20: The Opus Card page to unlink opus card. For that user needs to click on trash icon to delete particular opus card.

Card has been unlinked Successfully.

Id	Number	Balance	Operation
3	7132819765986	0	

Figure 5.21: The page showing success message that the opus card has been successfully unlinked from the account.

5.7 User story 9: View OPUS Card Balance

The existing iGo User can see balance for each linked opus card to his/her account.

Manage OPUS Cards			
Id	Number	Balance	Operation
2	12121212121212	10	

Figure 5.22: The page to see OPUS Card details including its balance.

5.8 User story 10: Load OPUS Card

The existing iGo User wants to load his OPUS Card using online payment via VISA/MasterCard.

iGo Home Manage Opus Cards [niravjd@gmail.com](#)

Load OPUS Cards

Opus Card Details

Opus Card Number
7132819765986

Current Balance
100

Write your name in the right fields. Also write your imaginary card number.
By clicking CCV field card will turn.

Amount

Payment Information	
First Name	Surname
Card Number	
MM / YY	CCV

SUBMIT

Figure 5.23: The web page to load OPUS Card.

The screenshot shows a web application interface for managing OPUS cards. At the top, there is a navigation bar with links for 'iGo', 'Home', 'Manage Opus Cards', and a user account section. The main content area is titled 'Load OPUS Cards'. A box labeled 'Opus Card Details' displays the card number '7132819765986' and a current balance of '0'. Below this, a message instructs the user to write their name and card number, noting that clicking the CCV field will turn it on. The form fields include 'Amount' (which has a red border and a validation message 'Please fill out this field.'), 'First Name', 'Surname', 'Card Number', 'MM / YY', and 'CCV'. A large blue 'SUBMIT' button is at the bottom.

Figure 5.24: if user does not fill the certain information which is required to process payment, user is shown message to fill the required information first and then process.

The screenshot shows a 'Manage OPUS Cards' page. At the top, there is a navigation bar with links for 'iGo', 'Home', 'Manage Opus Cards', and a user account section. The main content area is titled 'Manage OPUS Cards'. A green banner at the top states 'Card has been loaded successfully.' Below this, there is a table with four columns: 'Id', 'Number', 'Balance', and 'Operation'. The table contains one row with the following data: Id 3, Number 7132819765986, Balance 100, and an 'Edit' icon in the Operation column. There is also a 'Delete' icon in the same row.

Figure 5.25: The message showing that card has been loaded and its updated balance.

Chapter 6

Team Contribution

Chapter 1 : Introduction	Nirav Patel,Rohan Deepak Paspallu,Jingya Pan, Divya Pandit,Koshaben Patel
Chapter 2: User Stories	Nirav Patel,Rohan Deepak Paspallu,Jingya Pan, Divya Pandit,Koshaben Patel
Chapter 3: Persona	Nirav Patel,Rohan Deepak Paspallu,Jingya Pan, Divya Pandit,Koshaben Patel
Chapter 4: Traceability Matrix	Nirav Patel,Rohan Deepak Paspallu,Jingya Pan, Divya Pandit,Koshaben Patel
Chapter 5: Implementation	Nirav Patel,Rohan Deepak Paspallu,Jingya Pan, Divya Pandit,Koshaben Patel

User Story	Suggested By	Implemented By
7, 8	Nirav	Rohan
3, 4, 10	Kosha	Nirav
5, 6	Jingya	Kosha
9	Divya	Jingya
1, 2	Rohan	Divya

Bibliography

- [1] Pankaj Kamthan. *Introduction to User Modeling*. 2019.
- [2] Pankaj Kamthan. *Traceability in Software Requirements*. 2019.
- [3] Pankaj Kamthan. *User Stories in Context*. 2019.
- [4] Pankaj Kamthan and Nazlie Shahmir. “A Characterization of Negative User Stories.” In: *SEKE*. 2016, pp. 579–582.

Glossary

Acceptance Tests These are tests conducted to determine if the requirements of a specification or contract are met . [5](#)

FLAG Flag variable is used as a signal in programming to let the program know that a certain condition has met.. [20](#)

FN Functional Requirement - defines a function of a system or its component, where a function is described as specification of behavior between outputs and inputs. [20](#)

GIGO Glossary for the iGo system . [3](#)

iGo The online Ticket Vending Machine Web Application integrating with STM system. [3](#)

Java Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible.. [4](#)

OPUS OPUS is the name of STM travelling card which used by people to travel by STM services, manufactured and distributed by STM agencies. [3](#)

Priority Priority is set for the user stories to check which need to be given top most priority so that we can break them down and start working on it immediately. [5](#)

REQ DESC Requirement Description. [20](#)

REQ ID Requirement ID. [20](#)

REQ SRC Requirement Source . [20](#)

REQ TYPE Requirement Type. [20](#)

REST API An application program interface that uses HTTP requests to GET, PUT, POST and DELETE data. [4](#)

Spring Framework The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.. [4](#)

STM Société de transport de Montréal (Montreal Transit Corporation). [3](#)

TMIGO Backwards traceability matrix of iGo. [20](#)

travellers People who use STM metros and buses to travel daily in Montreal, Quebec, Canada. [3](#)

user stories An informal, natural language description of one or more features of a software system
. [3](#)

USIGO User Stories of iGo. [4](#)