Concordia University

SOEN 6481 - Software Systems Requirements Specification

Deliverable 2

Requirement Analysis for Ticket Vending Machine

Supervisor
Prof. Pankaj Kamthan

Team A
Authors
Maria Ahmed - 40070844
Adarsh Aravind - 40082585
Sri Akhil Varma Alluri - 40082333
Dheeraj Ashok Shobha - 40082192
Charles Jebalitherson Augustin Moses - 40084105

Contents

Chapter 1

Problem 5

1.1 User Stories

1.1.1 Global Constraints

Global constraints are applicable on the whole iGo system at every point in time.

ID:	Constraints
G1	End user is expected to use iGo system by modern web browsers such as Google Chrome, FireFox, Microsoft Edge, Safari because iGo is developed as a Web Application. It is supposed to work efficiently with modern web browsers.
G2	iGo system is developed to connect with STM system. 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, developed or tested by iGo team.
G3	iGo does not support offline usage, therefore an iGo user is supposed to have a good internet connection when using the web application.

1.1.2 Local Constraints

Local constraints are specific to user stories. we have linked each local constraints with it's user stories in following sections.

ID:	Constraints
C1	User should use active email address to login or sign up into iGo system.
C2	User should use valid Canadian phone number to sign up into iGo system.
C3	User should have an address in Montreal for delivery of opus card.
C4	User will not be able to access old data if user reactivate account with same email address or phone number.
C5	User must have at-least one transaction against linked OPUS card

We have used MoSCoW method to prioritize the user stories and Fibonacci sequence to calculate the story points.

1.1.3 iGo Registration using E-mail ID/ Phone number:

User Story ID:	US-01
Description	As a new user, I want to sign-up to an iGo account using my email address or phone number, so that I can make online transactions
Local Constraints	C1,C2
Priority:	Must have
Story points:	5

Acceptance tests

ID	Pre-condition	Scenario	Output
AT-01.1	Valid e-mail address/ phone number and password.	User fills the registration form on iGo website to create an account.	• iGo validates e-mail id/ phone number and password. • After successful validation, iGo displays the home page to the user.

AT-01.2	Invalid e-mail id/ phone number and password	User fills the registration form on iGo website to create an account.	• iGo validates the email id and is invalid. • A message is shown to the user which states email address is not valid. • Account has not been created.
AT-01.3	Valid e-mail id/ phone number which is already in use.	User fills the registration form on iGo website to create an account.	• iGo validates the email id/ phone number and notifies the user that the email id already exists in the system. • Account has not been created.

1.1.4 Login to view user profile

User Story ID:	US-02
Description	As a registered user, I want to login to my iGo account, so that I can view my profile.
Local Constraints	C2
Priority:	Must have.
Story points:	7.

Acceptance tests

ID	Pre-Condition	Scenario	Output
AT-02.1	User must be a registered user and is at the login screen.	• User enters the username. • User enters the password. • User clicks on login.	The user can navigate to the profile page to view his profile.
AT-02.2	User must be a registered user and is at the login screen.	• User enters the username. • User enters the wrong password. • User clicks on login.	iGo displays a error message that says, the user did not enter valid credentials.
AT-02.3	User is a registered user and is logged in to iGo	User clicks on profile tab	iGo displays user profile: • User is able to view his profile picture along with is full name. • User is able to view the card linked to the account. • User is able to view his residential address. • User is able to view his contact information.

1.1.5 View Transaction History

User Story ID:	US-03
Description	As a registered user, I want to view the transaction history so that I can access the details such as time stamps, total cost of transaction and OPUS card used.
Local Constraints	C5
Priority:	Must have.
Story points:	5

Acceptance tests

ID	Pre-Condition	Scenario	Output
AT-03.1	User logged into iGo system to view transaction history	User clicks on view transaction history menu with following filter: • Choose the type of purchase to filter the transactions	iGo displays the result set based on options selected by user
AT-03.2	User has zero transactions on their account	User can choose to view: • Last 10 transactions	iGo displays an error message that says, the user did not perform any transactions so far.
AT-03.3	User has made transactions with a OPUS card for the last 6 months	User can choose to view • Last week transactions	iGo shows last week's transaction history with timestamps of the transaction, along with ticket price and type.
AT-03.4	User has made transactions for the last 3 weeks with a particular OPUS card that is linked with their iGo account	User can choose to: view last 6 months transactions.	iGo now displays last 6 months' transaction history with the day and time the transaction was made, along with ticket price and type.

1.1.6 Make Payment

User Story ID:	US-04
Description	As a registered user, I want to make a payment using debit/credit card.
Local Constraints	C3
Priority:	Must have.
Story points:	9.

Acceptance tests

ID	Pre-condition	Scenario	Output
AT-04.1	User has an active debit or credit card to make the payment.	• User clicks on the payment button after selecting the type of ticket he/she wants to purchase. • User enters the debit or credit card number, card holder name, CVV, Expiry date and postal code • Clicks on next button to proceed with the payment	'Payment successful' message will be displayed to the user

AT-04.2	User has an active debit or credit card to make the payment.	• User clicks on the payment button after selecting the type of ticket he/she wants to purchase. • User enters an invalid debit or credit card number(not equal to 16 digits). • Clicks on next button to proceed with the payment	User will be notified that card number must have 16 digits ; Next button will be disabled and user will not be able to make payment
AT-04.3	User has an active debit or credit card to make the payment.	• User clicks on the payment button after selecting the type of ticket he/ she wants to purchase. • User enters an invalid Expiry date. • Clicks on next to proceed with the payment	User will be notified to enter a valid expiry date; Next button will be disabled and user will not be able to make payment

AT-04.4	User has an active credit/debit card to make the payment.	• User clicks on the payment button after selecting the type of ticket he/she wants to purchase. • User enters invalid CVV(not equal to 3 digits). • Clicks on next to proceed with the payment	User will be notified to enter a valid CVV; Next button will be disabled and user will not be able to make payment

1.1.7 Ticket Purchase of Guest User:

User Story ID:	US-05
Description	As a guest user, I want to purchase tickets without logging into the system.
Priority:	Must have.
Story points:	3.

Acceptance tests

ID	Pre-condition	Scenario	Output
AT-05.1	Guest user does not have to login.	• Guest user is at the start screen of iGo • Guest user does not enter login credentials. • Guest user clicks on guest user option.	iGo navigates to the guest user page and displays the purchase ticket option.

AT-05.2	Guest user clicks on guest user option.	• Guest user is in guest user screen. • Guest user clicks on purchase ticket option	The screen presents options to select the type of ticket. Day pass, one way, week pass, weekend pass.
AT-05.3	Guest user clicks on guest user option	• Guest user is at the purchase ticket option screen • Guest user is browsing the types of tickets.	• iGo does not display option for recharging OPUS card.
AT-05.4	Guest user is browsing the ticket options	• Guest user selects the type of ticket and enters the details for payment.	• iGo shows ”purchase successful” message confirmation.

1.1.8 Renewal of Opus card:

User Story ID:	US-06
Description	As a registered user, I want to place an order to renew my OPUS card.
Local Constraints	C3
Priority:	Could have.
Story points:	3.

Acceptance tests

ID	Pre-condition	Scenario	Output
AT-06.1	Registered user is logged in to his/her account.	User clicks ”Renew Opus Card” button • Message ”Are your current address and contact information up to date?” appears • User clicks ”Yes.”	Message “Order is placed. You will get your new opus card within 7 business days.”

1.1.9 Deactivate User Account:

User Story ID:	US-07
Description	As a registered user I want to deactivate my user account from iGo system so that all my personal information including transaction history can not be accessed from iGo.
Local Constraints	C4
Priority:	Should have.
Story points:	5.

Acceptance tests

ID	Pre-condition	Scenario	Output
AT-07.1	Registered user is logged in to his/her account.	User clicks ”deactivate account” button • Message box to re-enter user name and password appears to verify the user • User re-enters right user name and password. • User clicks verify button on the message box. • Message “You will no longer be able to use this account or view your transaction and payment history. Are you sure you want to delete your account?” appears • User clicks ”Yes.”	Message “Your account has been deactivated.” will appear and the system will be logged out automatically.
AT-07.2	Registered user enters wrong username or password in user verification box.	User clicks verify button on the message box.	Message box with “Sorry username or password is incorrect” will appear and the system will be logged out automatically.
AT-07.3	A user tries to login with deactivated username and password.	User clicks login button.	Message “Sorry the username or password is incorrect.” will appear.

1.1.10 Negative User Stories:

NUS-01: As a fraudulent I want to use someone else's debit/credit card to purchase ticket from iGo.
In iGo system user does not save registered users's debit/credit card information. In case anyone hack into iGo system they will not be able to use saved debit/credit card information for any payment.

NUS-02: As an identity thief I want to use someone else's account to login so that I can access iGo services.
Identity thief can target and misuse an account that is not in use for long time. If an user want to deactivate his/her account at any time they can do it. This service will help to protect users identity who no longer use public transport frequently or who may be moved from Montreal city permanently.

1.2 Quality attributes

Security: Security is provided in the system by authenticating only registered users. On the other hand, guest users can login to the system without authentication but can access only specific pages such as home page and ticket purchase page. Payments are securely handled by financial institutions with high data security standards.

Usability: User interface of our application is simple and easy to navigate across multiple web pages. All these pages are grouped together under common menu, which provides an abstract view of all the sub topics and provides pointers to locate the appropriate information.

Accessibility: Our system will be accessible to most of the users utilizing the system in mobile and computers. But, visually challenged users will need assistance to use the application

Sustainability: We have achieved the sustainability with the help of following techniques namely interviews, project reviews and collaborative user stories. A group of semi-structured interviews collected from different stakeholders of system at different stages of development to accommodate modification to the system thereby achieving sustainability. A collaborative user story building helps us to instantiate the sustainability goal model and the corresponding quality model, along with analyzing and documenting sustainability user stories and if relevant, also experimenting with the sustainability recommended system.

Atomic & Consistent: All our user stories are atomic in nature with a standard structure shown below:
As a $< role >$, I want $< goal >$ so that $< operation >$

From the above example, it is clear that the user stories are written with clear purpose in atomic manner where it serves one goal at any given time. The user stories are arranged in work flow order of application. This creates a smooth transition from one user story to another which helps us to achieve consistency

Validatable & Verifiable: Each user story has its own acceptance tests associated with it. Using the acceptance tests, we can perform acceptance testing where we can validate and verify the user story whether the system satisfies all the test cases written per user story.

Portability: Our application uses HTML, CSS, JS, JAVA to implement the server side and client side business logic, validation and control flow of the application. All these technologies are open source and widely used across different platforms. Hence, migrating the application from platform to another won't be difficult and requires minimal work.

Ambiguity: The user stories are defined such that there is no chance for any kind of ambiguity to occur.

Indeterminacy: It's taken care to avoid vagueness and generality in the user stories by providing clarity of the user , his action and the goal that he intends to achieve.

1.3 Personas

1.3.1 Guest user

Figure 1.1: Alexa Bezzos

Name: Alexa Bezzos
Demographics
Age: 26
Status: Single
Location: Montreal
Occupation: Marketing Director

Ambitions: To expand her skills, Build her own blog.

Transit Bio: Alexa usually works from home. But sometimes will go to office and she will have to leave the house during the rush hour.

Frustrations:

1. Standing in long queues
2. Not able to buy iGo transit ticket remotely on her phone

Goals:

1. Go to the guest user option
2. Buys a two way ticket

1.3.2 Registered user

Figure 1.2: Samuel Logan

Name: Samuel Logan

Demographics
Age: 23
Status: Single
Location: Montreal
Occupation: Student

Ambitions: To travel the world.
Transit Bio: Samuel usually goes to University 5 days a week. He also visits a lot of different places often.

Frustrations:

1. Wasting time to recharge opus card at the beginning of month.
2. Forgetting to re-new opus card.
3. Not able to see travel history.

Goals:

1. Registers to iGo system.
2. Recharges and re-news opus card from iGo system.
3. Access travel history from iGo.

Chapter 2

Problem 6

2.1 Traceability matrix

User Story	**Source**	**Implementation**	**Acceptance test**
US-01: iGo registration	S1:D1 Description of iGo S2:D1 Use Case model of iGo	I-01	AT-01.1 to AT-01.3
US-02: View user profile	S1:D1 Description of iGo S2:D1 Use Case model of iGo	I-02	AT-02.1 to AT-02.3
US-03: View Transaction History	S1:D1 Description of iGo S2:D1 Domain model of iGo S3:D1 Use case model of iGo S4:D1 Sequence diagram of iGo S5:D2 User Story 2	I-03	AT-03.1 to AT-03.4
US-04: Payment	S1:D1 Description of iGo S2:D1 Domain model of iGo S3:D1 Use case model of iGo	I-04	AT-04.1 to AT-04.4
US-05: Ticket Purchase of guest user	S1:D1 Description of iGo S2:D1 Domain model of iGo S3:D1 Use case model of iGo	I-05	AT-05.1
US-06: Renewal of OPUS card	S1:D1 Description of iGo S2:D1 Use case model of iGo S3:D1 Sequence diagram of iGo	-	AT-06.1
US-07: Deactivate user account	S1:D1 Description of iGo	-	AT-07.1 to AT-07.3

Chapter 3

Problem 7

3.1 User Story Implementation

3.1.1 I-01: Signup

US-01:
User story written by: Adarsh Aravind
User story implemented by: Maria Ahmed
Implementation is done using HTML,CSS and javascrit. All codes are provided inside -01 folder.

1. Go inside I-01 folder.
2. Click signup.html file.

AT-01.1: First Name, Last Name and Password are mandatory fields, user can choose one field between email and phone number and fill one or both.Minimum length of the password must be 6 and password must be a combination of both number and alphabets. Then click submit button. If the email and/or phone number is valid and never used before for iGo, user should be able to sign up. If sign up is successful page should redirect user to profile page (Figure:3.5 Profile page).

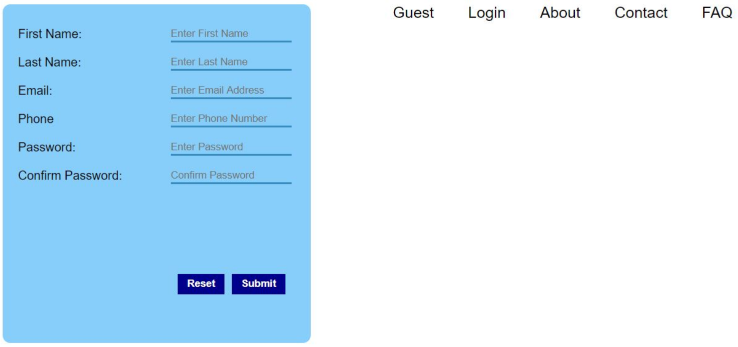
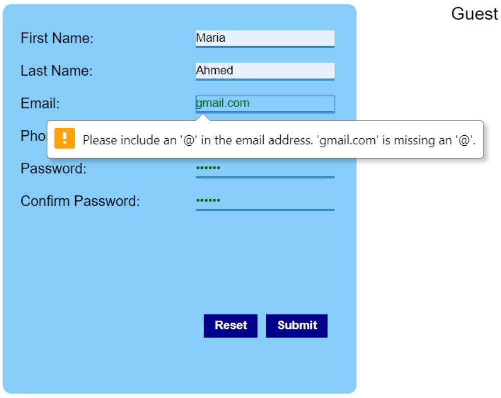


Figure 3.1: Sign up form

AT-01.2:Provided email address must have correct format otherwise signup form will show error message.

AT-01.3: If provided email has been used already to make account in iGo then system will through ”email already exists.” message.

If provided phone number has been used already to make an account in iGo then system will



First Name: Maria

Last Name: Ahmed

Email: gmail.com

Pho

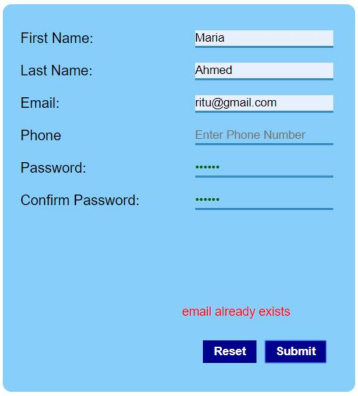
Please include an '@' in the email address. 'gmail.com' is missing an '@'.

Password: ••••••

Confirm Password: ••••••

Figure 3.2: wrong email

through ”phone number already exists.” message.
To check this acceptance test use ritu@gmail.com as email and 438234567 as the phone number.



First Name: Maria

Last Name: Ahmed

Email: ritu@gmail.com

Phone Enter Phone Number

Password: ••••••

Confirm Password: ••••••

email already exists

Figure 3.3: email exists

3.1.2 I-02: Login to iGo and view profile, transaction history

User story written by: Dheeraj Ashok Shobha
Implemented by: Adarsh Aravind
The user story has been implemented using HTML and CSS.
Find the code in the folder named I-02. Double click on index.html which opens the login page. Username can be anything, but the password is hard-coded to abcd To run the program, please follow the below instructions:

1. Open index.html page and enter username, password. Click on Login button. If you forget your password or if you don’t have an account, there are two options. (“Forget password?” and “Don’t have an account?”)

   (a) Forget password redirects you to resetpassword.html page where you can provide your details again to create new password.

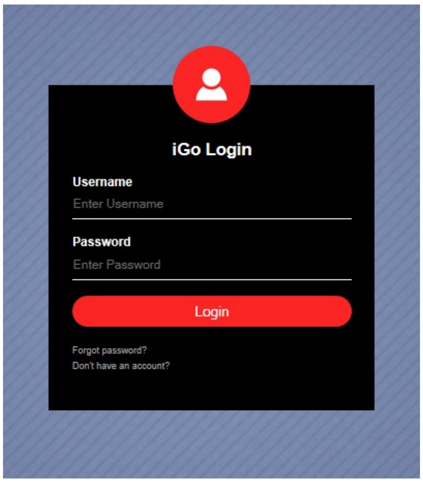
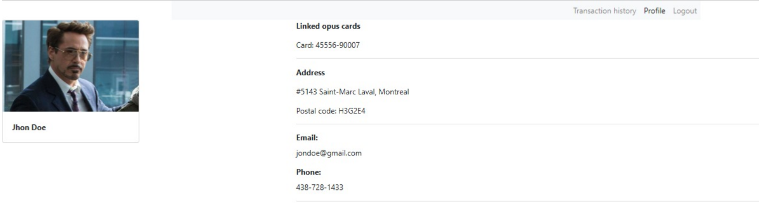


Figure 3.4: login form

(b) If you click on "Don't have an account" you will be redirected to signup page, where you can create a new account by entering your details in the form. Once entering the details, click on "Sign Up" button, which will again redirect to the login page.

2. Now, enter your details and click on Login button. It will redirect you to profilepage.html where you can check your card information, and personal details related to the iGo account.



Jhon Doe

Linked opus cards

Card: 45556-90007

Address

#5143 Saint-Marc Laval, Montreal

Postal code: H3G2E4

Email:

jondoe@gmail.com

Phone:

438-728-1433

Figure 3.5: Profile page

3. On the same page, you have 3 options at the top namely Transaction History, Profile, and Logout.

   (a) If you click on Transaction History, you will be redirected to transactionhistory.html page where you can view previous transaction information.

   (b) If you want to view your transaction history and view your profile information again, click on profile button.

   (c) If you click on Logout button, you'll be redirected to iGo login page.

This implementation satisfies all the acceptance test cases written for US-02 (Login to iGo account and view profile).

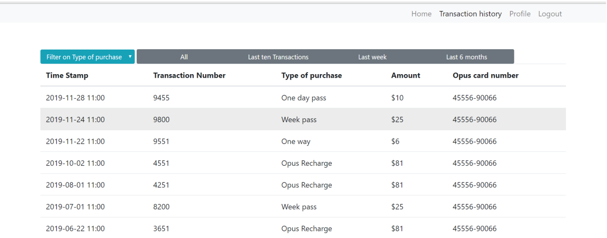
To view the implementation, **Click on:** Login implementation
or copy the url to a browser: https://userstory02-i02.s3-sa-east-1.amazonaws.com/index.html

All the code files have been uploaded on the submission folder.

3.1.3 I-03: View Transaction History

User story written by: Charles Jebalitherson Augustin Moses
Implemented by: Dheeraj Ashok Shobha

The user story has been implemented using HTML, Bootstrap and javascript.
The user has been given the option to view the transaction history based on 3 filters of time range (i.e last 10 transactions, last week, last 6 months), he can also filter the transactions based on the type of purchase (click on the drop down filter option)



Time Stamp	Transaction Number	Type of purchase	Amount	Opus card number
2019-11-28 11:00	9455	One day pass	$10	45556-90066
2019-11-24 11:00	9800	Week pass	$25	45556-90066
2019-11-22 11:00	9551	One way	$6	45556-90066
2019-10-02 11:00	4551	Opus Recharge	$81	45556-90066
2019-08-01 11:00	4251	Opus Recharge	$81	45556-90066
2019-07-01 11:00	8200	Week pass	$25	45556-90066
2019-06-22 11:00	3651	Opus Recharge	$81	45556-90066

Figure 3.6: transaction history

AT-0.32 has not been satisfied by this implementation as the pre-condition is that the user should have zero transactions as this implementation does not have a database the values are hard coded transaction histories hence they can't be zero.

Instructions to run the user story implementation: **Click on:** Transaction history implementation
or copy the url to a browser: https://userstory3-i03.s3-sa-east-1.amazonaws.com/index.html All the code files have been uploaded on the submission folder.

3.1.4 I-04: Make Payment

US-04: As a registered user, I want to make a payment using debit/credit card.

Written by: Akhil
Implemented by: Charles Jebalitherson

This user story has been implemented using Angular JS, SCSS, HTML, Bootstrap and Firebase.

The live application is hosted in the url: `https://igo-web-app.firebaseapp.com`

Working Model

Using the above url, we can access the landing page where we can select the type of ticket. After the ticket selection and clicking on next button. It will be redirect to payment page where user needs to enter the card details and postal code.

Validation:

Payment page performs following validations before making a payment:

- All the fields are required

- Card number must contains 16 digits and accepts only numbers
- CVV must have 3 digits and accepts only digits
- Postal code must be alpha-numeric and length of 6.

The Next button in payment page gets enabled only after all the above validations are successful. After successful validations, user can click on next button which shows a pop-up ”Amount to be paid for the ticket”. Once the user clicks on ”Confirm Payment”, user will get ”Payment Successful” message.

This implementation satisfies all the acceptance test cases written for payment user story.

Instructions to run the application locally

- Install Node JS (https://nodejs.org/en/)
- Install Angular CLI. Use command npm install -g @angular/cli
- Run the command ng serve –open in the project folder I-03
- Navigate to the url http://localhost:4200/ to view the application

3.1.5 I-05: Ticket Purchase of guest User

US-05: As a guest user, I want to purchase a ticket without logging into the system.

Written by: Dheeraj Ashok Shobha
Implemented by: Sri Akhil Varma Alluri

This user story has been implemented using Java, JFrame

The live application is in the executable .jar file (i-05-executable) which is present in the folder that is shared.

Working Model

Using the above mentioned .jar file, we can access the main page where we can select the guest user option which navigates to a new page in which we need to enter the details specified. After entering all the details click on the purchase ticket option. After clicking on the purchase ticket option. It will display a box with message ticket purchase successful.

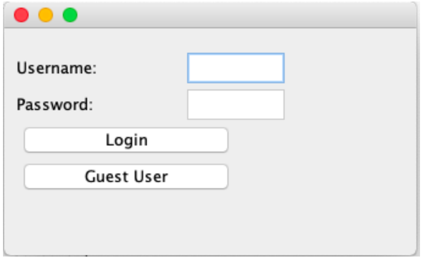
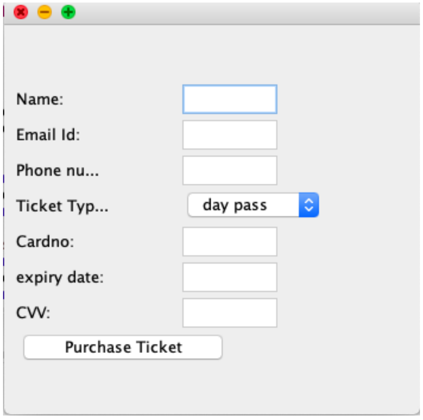


Figure 3.7: GuestUser Interface



Name:

Email Id:

Phone nu...

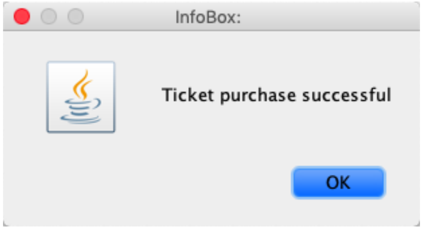
Ticket Typ... day pass

Cardno:

expiry date:

CVV:

Figure 3.8: Ticket Purchase Interface



Ticket purchase successful

Figure 3.9: success message

Validation:

This implementation satisfies all the acceptance test cases written for Ticket purchase of Guest user story.

Instructions to run the application locally

- Install Eclipse (https://www.eclipse.org/downloads/)
- Install Java 1.8
- (https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

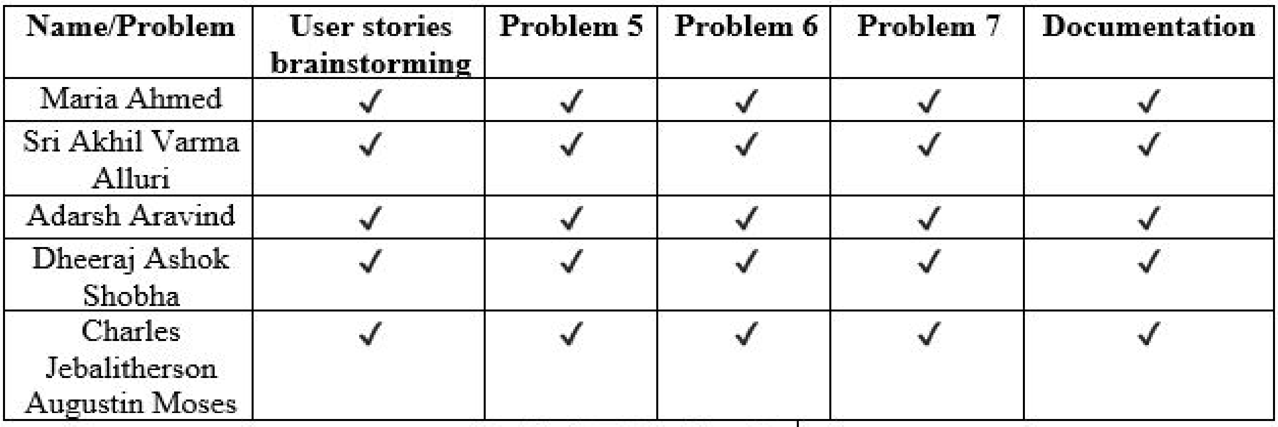
Chapter 4

Glossary

Guest User	A user who doesn't have a user account with iGo.
Registered User	A user who has a user account with iGo
MoSCoW method	The MoSCoW method is a prioritization technique.
Fibonacci	In mathematics, the Fibonacci numbers, commonly denoted Fn form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1.
US	User Story.
AT	Acceptance test.
I	Implementation.
iGo	TVM system that is being developed.

Chapter 5

Work Allocation



Name/Problem	User stories brainstorming	Problem 5	Problem 6	Problem 7	Documentation
Maria Ahmed	✓	✓	✓	✓	✓
Sri Akhil Varma Alluri	✓	✓	✓	✓	✓
Adarsh Aravind	✓	✓	✓	✓	✓
Dheeraj Ashok Shobha	✓	✓	✓	✓	✓
Charles Jebalitherson Augustin Moses	✓	✓	✓	✓	✓

Table 1: Task allocation

Chapter 6

References

1. https://users.encs.concordia.ca/~kamthan/courses/soen-6481/software_product_quality_introduction.pdf
2. https://users.encs.concordia.ca/~kamthan/courses/soen-6481/clear.pdf
3. https://users.encs.concordia.ca/~kamthan/courses/soen-6481/software_requirements_traceability.pdf
4. https://users.encs.concordia.ca/~kamthan/courses/soen-6481/user_stories_context.pdf
5. https://cli.angular.io/
6. https://getbootstrap.com/docs/4.4/getting-started/introduction/
7. https://alligator.io/angular/deploying-angular-app-to-firebase/
8. “Password Validation” https://www.w3schools.com/howto/howto_js_password_validation.asp
9. “Validate a password using html and JS” https://www.geeksforgeeks.org/validate-a-password-usi
10. “Sign up form” https://www.w3schools.com/howto/howto_css_signup_form.asp
11. “How to make Login form”, Easy Tutorials https://www.youtube.com/watch?v=0WNxUVnY3pg