

# SOEN 6481 - Software System Requirements Specification

Prof. Pankaj Kamthan

Team G

Concordia University  
Department of Computer Science and Software Engineering  
Montreal

November 29, 2019

# STM Ticket Vending Machine - Existing system problems

## Problem 1: Busy Schedule Makes it all Tedious and Cumbersome.

- People seem to be having so busy life that standing in long queues during peak hours to get tickets for travel seems to be hectic.

"I think that I was a rat in rat wheel in my previous life... can not forget the habit"  
— Hiroko Sakai



(a)



(b)

## STM - Existing system problems

### Problem 2: Not portable , thus a time consuming process

- There are machines installed on stations for purchasing tickets which seems to be less cost efficient, less space efficient and not portable at all.



# iGo system - new Approach , new vision!

## Benefit 1 : Anywhere ,Anytime and Easy Access

- iGo is an online application,increases the conveniences to operate anywhere and anytime.It needs less maintenance and adds security to system.

## iGo : Ticket Vending Machine System

Riding Together Make it Better !!

USER LOGIN

(c)



(d)

# iGo system - new Approach , new vision!

## Benefit 2 : Go Digital, Go paperless

- iGo system is completely paperless as its a web application successful purchase of ticket will be notified by email or sms.



(e)



(f)

# iGo system - new Approach , new vision!

## Benefit 4 : Digital Payment ,Say no to cash!

- iGo system only accepts online payments with Visa/Credit/debit or Interac .Thus, making society cashless.



(g)



(h)



(i)

# User Story 1

- An existing iGo user can create a new account with unique email, however appropriate error message is displayed stating relevant information i.e."Email id is already registered".

The screenshot shows a registration form titled "User Registration" on a web page. The URL in the address bar is "localhost:8080/registerUser". The form contains six input fields: "Email Id", "Password", "User First Name", "User Last Name", "Address", and "Phone Number". Below the form are two buttons: "REGISTER" and "RESET". A green label "(j)" is positioned below the "REGISTER" button.

localhost:8080/registerUser

### User Registration

Email Id

Password

User First Name

User Last Name

Address

Phone Number

REGISTER    RESET

(j)

Figure: Registration form of iGo

## User Story 1 Contd.

- An existing iGo user can create a new account with unique email, however appropriate error message is displayed stating relevant information i.e."Email id is already registered".

The screenshot shows a web browser window with the URL `localhost:8080/userRegistered`. The main content is a registration form titled "User Registration". The "Email Id" field contains `divyapandit13@gmail.com`. A red error message "This Email is already registered in the system!!" is displayed prominently above the form. Below the message, the form fields are as follows:

Email Id	<code>divyapandit13@gmail.com</code>
Password	<input type="password"/>
User First Name	<code>Home</code>
User Last Name	<code>Last Name</code>
Address	<code>936 Rue De l'Eglise, Apt 4</code>
Phone Number	<code>5145591396</code>

At the bottom of the form are two buttons: "REGISTER" and "RESET".

(a)

Figure: Email id exists in iGo database already

## User Story 2

An existing iGo user login iGo website by providing credentials to use services offered by iGo.



Please Sign in

Email or Password invalid, please verify

[Forgot Password?](#)

© 2019-2020

Figure: Login form of iGo

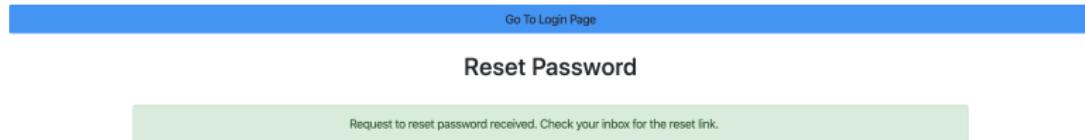
## User Story 3

An existing iGo user can reset my password so that he/she can get back in to iGo site if he/she forgets my password.

The screenshot shows a web page titled "Reset Password". At the top right is a blue button labeled "Go To Login Page". Below the title is a text input field with the placeholder "Email" and a label "Email" above it. At the bottom of the input field are two buttons: "Clear" (gray) and "Reset Password" (blue).

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

## User Story 3 Contd.



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

## User Story 4

An existing iGo user having physical OPUS card can link OPUS card to his/her iGo account after he/she logs into the iGo system.

The screenshot shows a web application interface for linking an OPUS card to an iGo account. At the top, there is a dark header bar with the text "iGo Home Manage Opus Cards" on the left and "niravjd@gmail.com" with a dropdown arrow on the right. Below the header, the main content area has a title "Link OPUS Card" centered above a light gray rectangular form. Inside the form, there is a label "Opus Card Number" positioned above a white input field. Below the input field are two buttons: "Clear" on the left and "Submit" on the right, both in a dark blue color. The entire form is set against a light gray background.

**Figure:** 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.

## User Story 5, 6

An existing iGo user can remove a linked OPUS card from his/her iGo account, to delete my information from iGo system.

An existing iGo user can see remaining balance in his/her linked opus cards.

The screenshot shows a web application interface titled "Manage OPUS Cards". At the top, there is a navigation bar with links for "iGo", "Home", "Manage Opus Cards", and a user email "niravjd@gmail.com". The main content area is titled "Manage OPUS Cards". Below the title is a table with four columns: "Id", "Number", "Balance", and "Operation". A single row is present in the table, showing "2" in the Id column, "1212121212126" in the Number column, "10" in the Balance column, and two icons in the Operation column: a trash bin and a clipboard.

Id	Number	Balance	Operation
2	1212121212126	10	

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

## User Story 7

An existing iGo user having at least one linked OPUS card, can top up his/her particular linked OPUS card on iGo using visa/mastercard.

The screenshot shows a web browser interface for the iGo platform. 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 "Load OPUS Cards". A box titled "Opus Card Details" contains the card number "7132819765986" and the current balance "100". Below this, a note says "Write your name in the right fields. Also write your imaginary card number. By clicking CCV field card will turn." An "Amount" input field is present. A large blue "Payment Information" section contains fields for "First Name" and "Surname" (with a small "i" icon), "Card Number", "MM / YY", and "CCV". A "SUBMIT" button is located at the bottom of this section.

Figure: The web page to load OPUS Card.

# Persona modeling

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.

**Quote**  
“Every problem has a solution.”

**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. Likes playing football, computer games. Likes to attend social events.

**Typical Usage**

Internet	Mobile device
85	80

Banking Applications
80

**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, usernames
- Animations
- Advertisements
- Excessive features
- Having to wait in long queue to recharge opus card

# Persona modeling

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."  
"

**Motivations**

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

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

**Typical Usage**

Internet
85

Mobile device
85

Banking Applications
85

**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

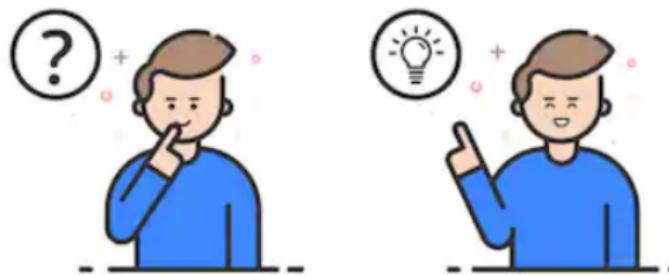
**Frustrations**

- 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

# Collaboration Patterns

## Pattern 1 : Clear Up Questions

- We kept meeting to clear our doubts within the project. Every member could come up with some unique information and plans for the work to be done.



(c)

# Collaboration Patterns

## Pattern 2 : Share Expectations

- The meeting would end up being long discussions on problems or risks that would be faced while allocating certain tasks. every person would explain what goals needs to be achieved by every week in-order to complete the deliverable with ease and perfection.



(d)

# Collaboration Patterns

## Pattern 3 : Fill Knowledge Gaps

- If some member would face certain issue we would sit back and try solving the issue together. We did in deliverable 2 also team member n reviews source code of team member n+1 for filling up gaps and improvements in results.



(e)

# Collaboration Patterns

## Pattern 4 : Centralize Work Product Management

- Ensure that all team members always have access to the latest project artifact versions. thus we kept pushing the most recent code on GitHub for everyone's access

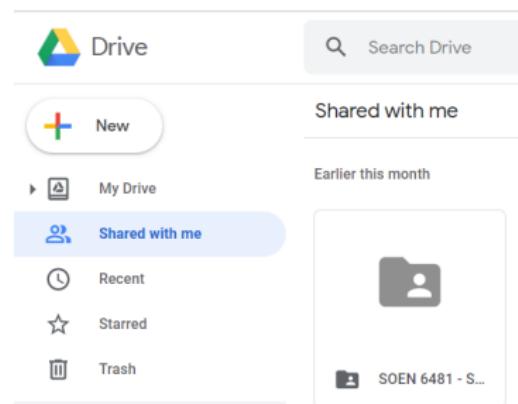
The screenshot shows a GitHub repository page for 'niravjdn / SRS-Project'. The repository has 48 commits, 1 branch, 0 packages, 1 release, and 5 contributors. The latest commit was made yesterday. The repository contains files like 'Deliverable 2', 'Deliverable1', 'Interview Recordings', 'README.md', and 'TeamG-Deliverable1.zip'. A pull request button is visible.

File	Commit Message	Time
Deliverable 2	Update README.md	yesterday
Deliverable1	Modified Live Demo Link	11 days ago
Interview Recordings	Interview of TA (Samia Hilal)	13 days ago
README.md	Update README.md	5 days ago
TeamG-Deliverable1.zip	Added D1	2 months ago

# Collaboration Patterns

## Pattern 5 : Regularly Check Requirement Fulfillment

- we kept checking with the teaching assistant and professor on what is expected in particular deliverable. For team members we always used the drive to update every possible artifact used for the project.

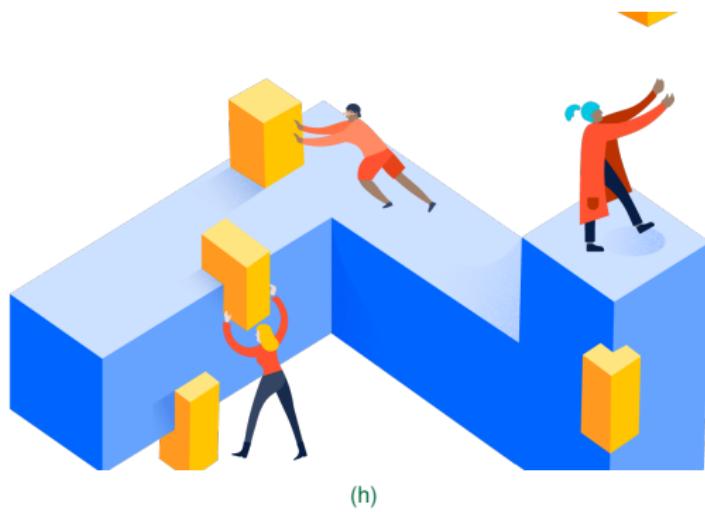


(g)

# Collaboration Patterns

## Pattern 6 : Manage The Project

- Appropriate roles, tasks, and responsibilities were always mutually decided for managing our project. And if at all one faces problems others were always ready to help that person.



# Critical Decisions



Figure: Spring Framework



Figure: Yii Framework

## Spring

- Java
- Open Source
- Great Community
- Matured
- More option for customization

## Yii

- Code Generator
- Active Record
- Good for small scale applications
- Simple
- Limited Option for Customization

## Potential Re-use of IGo

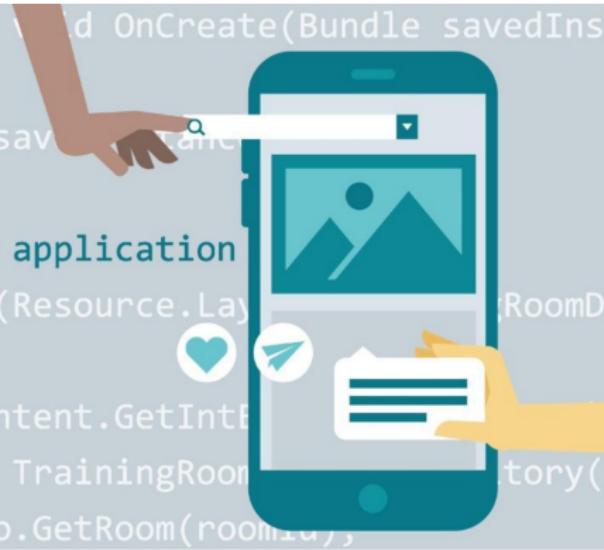
This web based application can be used across any online ticketing system with a few more advancements in terms of criticality.



Code Reusability

## Potential Re-use of IGo

Interviews, user stories, personas and other documentations done during the project could be of significant use as they contain vital information of daily users of the system which could be further used to design more advanced systems globally.



```
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);
    // Create your application
    SetContentView(Resource.Layout.RoomDetail);
    int roomId = Intent.GetIntExtra(Intent.EXTRA_ROOM_ID, 0);
    var repo = new TrainingRoomRepository();
    var room = repo.GetRoom(roomId);
```

## Lessons learnt by doing project

- Importance of teamwork
- Empathizing with users
- Importance of understanding problem domain
- Deliver features needed by users
- Time Management
- Achieving Quality in one's work