November 25, 2023

ICT20015 - Assessment 4: Project Report

Trac Duc Anh Luong
103488117@student.swin.edu.au
Swinburne University of Technology

Name: Trac Duc Anh Luong - ID: 103488117

Contents

1. Introduction

OCG Technology JSC is a technology company founded in 2016 with the mission of providing multi-platform IT solutions for the benefit of the community and improving service quality for Vietnamese society. OCG offers outstanding education and application development services, including Smart Education solutions, software development services based on the OutSystems platform, and assistance to foreign partners connecting to and developing the system with a growing team of developers. NTT e-Asia is the company which is the biggest shareholder in OCG. NTT e-Asia is a subsidiary of NTT East Group Japan, which provides leading telecommunications and information technology services in Japan. In addition to Vietnam, NTT e-Asia also aims to expand business markets in Asian countries.

Every quarter of a year, the company organised an internship program consisting of software development training and application, intending to recruit new talented developers to be part of their official team of employees. After a 3-step application process of resume screening, technical test (including two parts: algorithms and SQL knowledge), and interview, I was officially recruited by the company to commence as an OutSystems Web Developer Intern from September 15 to November 15, 2023. As a final-year student enrolled in Swinburne University of Technology's internship semester, I aim to develop vital workplace skills and culture, including technical and soft skills. In addition, being an international ICT student also requires me to adhere to the ANZSCO Code Information [1], as it assists workers in nominating an ICT occupation in Australia and internationally.

2. Project Summary

2.1. Purpose

Employed as a Software Developer Intern, after a 2-week training session, my task is to develop a personal project, a fully functional reactive web application of my chosen topic. As OutSystems' personal environment's functionalities are extensive and offer services close to the enterprise environment, with only a few limitations related to performance and database storage, practising in the personal environment before accessing the development and testing workflow is very beneficial. The web application needs to follow a list of skill-specific requirements, which ensures that the team of interns can fully develop their understanding and skills of OutSystems, a new low-code web and mobile application development platform that was built on top of the ASP.NET framework and Microsoft SQL Server. During the development process, each intern was accompanied by experienced mentors in the field, who assisted and gave us feedback on our work for an efficient internship program.

2.2. Deliverables

After the 2-week training session, I received an outline of expected project deliverables. For the project's topic, I have chosen e-commerce, explicitly developing a website for an electronic store that contains product placement, order delivery and other data analytics functionalities for the system's admin. The website will have two separate roles, Customer and Admin, and business logic will also be divided into two flows to accompany these roles. The web application must adhere to standard business and technological requirements, which include good performance, device responsiveness, and error-prone experience for the customer. At the end of the internship, there would be two interviews: 1 for our final project, for which we will need to present the functionalities and requirement fulfilment, and 1 for technical skills, in which we would be interviewed about OutSystems knowledge, database, OOP, data structures, problems solving, and programming language-related skills that were included in our resume. After the interviews, the company's project managers and human resource team will determine if we are qualified for a return offer as a Fresher Outsytems Software Developer.

3. Project Requirements

There are a total of 24 requirements, divided into two categories: mandatory - requirements that are critical for the understanding of the OutSystems, and non-mandatory, requirements that are advanced and can help develop a deep knowledge of high-level functionalities of the platform but should only be taken into considerations after the interns have fulfilled all mandatory requirements. I have accomplished 19/24 criteria for my project, including 16/16 mandatory criteria and 3/8 non-mandatory and advanced criteria. Table 1 demonstrates the details of my project requirements.

No	Requirement	Level	Mandatory	Completed
1	Validate user input. Use built-in validation and custom validation.	1	Yes	☒
2	Use regular expression.	2	Yes	☒
3	One Time Password.	5	No	☒
4	Export data to Excel file.	1	Yes	☒
5	Export data to Excel file with format (e.g., add borders, merge cell, background colour, …)	3	No	☒
6	Import data from Excel file to entity.	1	Yes	☒
7	Import data from CSV file to entity.	3	No	□
8	Authorization for users.	1	Yes	☒
9	Write the user action log.	3	No	□
10	Use statistical graphs.	4	No	☒
11	Use modal.	2	Yes	☒
12	Use Site Property.	1	Yes	☒
13	Use Timer.	1	Yes	☒
14	Use Forge's components.	1	Yes	☒
15	Use Blocks.	2	Yes	☒
16	Display or hide element in the screen based on the user role.	1	Yes	☒
17	Use SQL Query.	1	Yes	☒
18	Send email.	1	Yes	☒
19	Create extension.	5	No	□
20	Connect to external database.	3	No	□
21	Comply with coding convention, naming convention.	1	Yes	☒
22	Use Business Process Technology.	5	No	□
23	Use CSS.	1	Yes	☒
24	Use JavaScript.	1	Yes	☒

Table 1: Project Requirements

4. Database Design

The database is one of the most essential parts of the application and its business logic process [2]. A good database design will help the development process be more seamless with CRUD operations and accurate query joins.

In Figure 1, you can see the tables in the application database. The database is effectively normalized by introducing weak entities as links between many to many relationships like Order and Product or Cart and Product. An innovative aspect of OutSystems is that it introduces static tables (or static entities), similar to enumerates in other programming languages like C# and Java. Status and Shipping are static entities with their Ids referenced by the Order entity. Status has five static records: Pending, Processing, Shipped, Completed, and Cancelled. Shipping has three static records: Express, Rush, and Economical. This assists both the Customer and Admin Roles manage orders more effectively without errors. There are also standalone static entities, including Months, AccountMenu, and HomeSort, which will be fetched from the database and displayed in dropdowns or radio lists for sorting and filtering in the application.

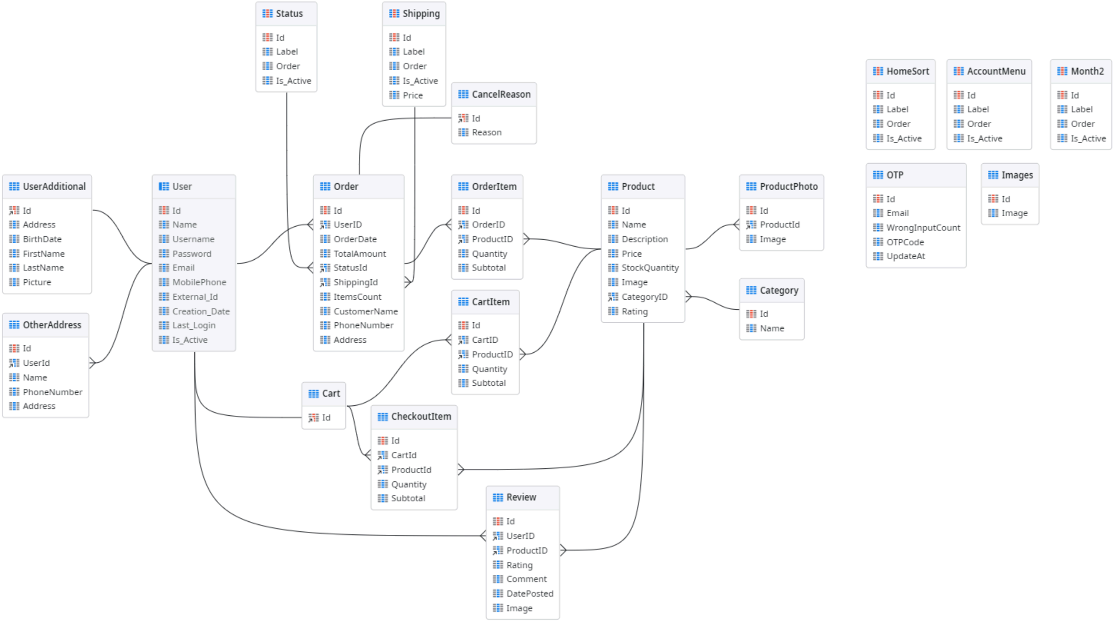


Figure 1: Data Model and Relationship

Key takeaways

The database design in the project serves as a foundation, facilitating a seamless development process with efficient CRUD operations. Normalization is employed effectively, introducing weak entities for many-to-many relationships and innovative static entities like Status and Shipping, enhancing order management with predefined options. These static entities, akin to enumerations in programming languages, minimize errors in order handling for customers and admins. Additionally, standalone static entities such as Months, AccountMenu, and HomeSort enhance user functionality by providing options for sorting and filtering through dropdowns or radio lists. The key takeaways from this database design process encompass gaining a solid foundation in analyzing business requirements, emphasizing the importance of accurate relationships for query joins, and showcasing the innovative use of static entities for improved user interactions.

5. User Flow

To visually represent the customer's interaction flow, I have created a UML diagram consisting of each screen's attributes and methods, which you can view in Figure 10 in the Appendices section. The flow consists of an e-commerce website's critical functionalities, including user authentication and OTP, product display, order placement and monitoring, product review and rating, user account management, SMTP (Simple Mail Transfer Protocol), and Automation.

5.1. User Authentication and OTP

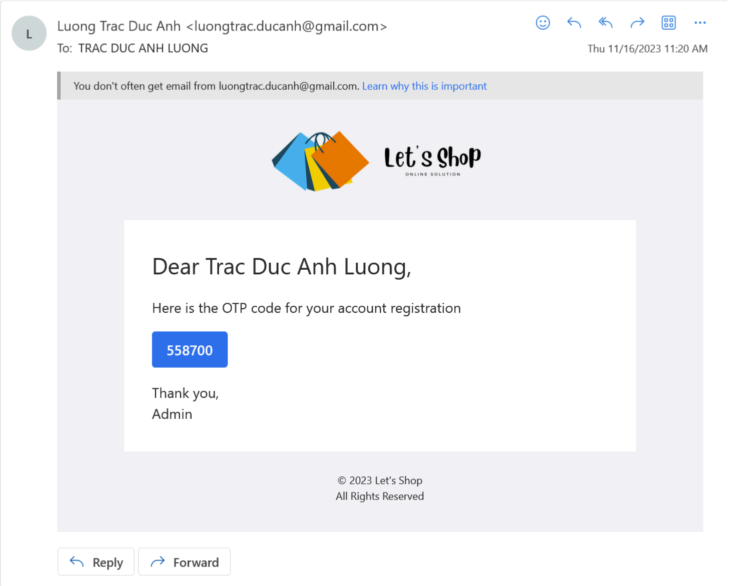
At their first access to the website, the users who are not yet registered can only view the product overview and product details. To access the website's main functionalities, users must log in to an existing account or register for a new one using their email as the primary authentication source.

In the login screen, the user can enter an existing account or sign up as a new user and be redirected to the register screen.

In registration, the user has to enter their details, including their name, email, address, mobile phone, date of birth, password, password confirmation, and optionally upload a profile picture. There are strict validation rules upon clicking the Register button, including:

a) Email must be unique and cannot be matched with any previously registered email in the database.
b) Mobile phones must comply with Vietnamese mobile format and be checked using JavaScript regex.
c) Date of birth cannot be in the future, and the user must be 18 years or older.
d) Password strength has to comply with at least:
   - 8 characters
   - 1 uppercase character
   - 1 lowercase character
   - 1 number
   - 1 special character
e) Confirm password and password must match.

The criteria above ensure the security of my application and avoid invalid data in the database. In addition, to guide the user during the password creation, I have added a responsive tooltip that will prompt when the user clicks on the password input and dynamically validate based on changes in the password field.



Luong Trac Duc Anh <luongtrac.ducanh@gmail.com>
To: TRAC DUC ANH LUONG
Thu 11/16/2023 11:20 AM

You don't often get email from luongtrac.ducanh@gmail.com. Learn why this is important

Dear Trac Duc Anh Luong,

Here is the OTP code for your account registration

558700

Thank you,
Admin

© 2023 Let's Shop
All Rights Reserved

Figure 2: OTP sent to user's email

After all input fields in the registration form are validated, the system will trigger actions to send the OTP (one-time password) to the user through their email (Figure 2) and start a 2-minute countdown. The OTP will only be valid before the countdown runs out, ensuring the security of the registration workflow.

The login page also has a hyperlink that users can click if they forget the password. The system will generate a new password string and send it to the user's email.

Key takeaways

The user authentication process exemplifies a commitment to robust security practices, utilizing email as the primary authentication source with stringent validation rules during registration. Password security is enhanced through guidelines necessitating a combination of characters and a responsive tooltip for real-time guidance. Incorporating a one-time password (OTP) system, sent via email with a countdown mechanism, adds an extra layer of security [3]. A user-friendly password recovery mechanism, activated by a clickable hyperlink on the login page, ensures seamless account access restoration. Compliance with localized mobile formats and age verification requirements demonstrates attention to regional standards and responsible user engagement. Dynamic validation, particularly the responsive tooltip during password creation, fosters transparency and guides users effectively.

5.2. Product Display

Three ways provide user access to view the shop's products. Each serves a different purpose of information display.

5.2.1. Home

The Home page is the entry point when users first access the website. This page shows running banners, product advertisements, and sections dedicated to best-rated products, and the product line is divided into various categories. The users can either view the product details or view groups of products by getting redirected to the Products page.

5.2.2. Products

The Products page is where all products are displayed. Highlighted functionalities of the page include:

a) Product categories filtering: Adding a variable `VarCategoryId` to the query and using the `or` operator `VarCategoryId = NullIdentifier() or Product.CategoryId = VarCategoryId`
b) Product sorting (Using a `switch` statement and a dynamic sort variable):
   - Name
   - Price - Low to High
   - Price - High to Low
   - Rating
c) Product search by name: Adding a `SearchKeyword` variable to the query and using the `or` operator `SearchKeyword = "" or Product.Name like "%" + SearchKeyword + "%"`
d) Pagination: 12 products are displayed on a page
e) Add to cart: The user can add an item and change the quantity on the Cart page.

5.2.3. Product Details

The Product Details page displays all the information related to a product, including name, category, description, rating, price, stock quantity, reviews, and similar products. You can adjust the selected quantity and add the item to your cart.

Key takeaways

Developing the Home, Products, and Product Details pages has enhanced my proficiency in dynamic SQL queries for efficient list filtering, sorting, and pagination. Integrating user interface (UI) and user experience (UX) principles [4], exemplified by strategic layout placement, responsive feedback, and seamless cart interactions, reflects a commitment to creating an engaging and user-friendly platform. The Products page, offering features like category filtering, versatile sorting options, and intuitive search functionality, underscores a user-centric approach. Attention to responsive design, visible in the Product Details page's adaptable quantity adjustments and seamless cart integration, ensures a consistent and enjoyable shopping experience across various devices. Additionally, the inclusion of feedback messages reflects a dedication to continuous improvement based on user interactions, reinforcing the platform's reliability.

5.3. Order Placement and Monitoring

Order creation and management are two of the most essential function flows of the application (You can view the flow in Figure 10). There are nested business process logic and edge case checking throughout this process. The users first add one or several items to the cart, which I used a Block to display. Block is a feature in OutSystems with a concept and functionalities similar to Components in JavaScript frameworks like ReactJS or VueJS. They are both reusable parts that can be implemented throughout the program. Modifying the Block will also change its instances, making the program consistent with minimal effort. The cart will display the number of items the user has previously added.

The users then go to the Cart page, adjust the item quantity, view the preview subtotal, and proceed to checkout. Users can select a shipping option on the Checkout page, keep or change their default address, view the final total price, and place an order. Upon the order placement, a set of data actions will run, including creating a new `Order`, adding `OrderItem` to the `Order`, removing `CheckoutItems` and `CartItems`, and sending an email with the order details to the user. Finally, the screen will be redirected to the Order Details page, where they can view the items in the order and their status. Optionally, the user can cancel the order before the status is Shipped and add the reason for their cancellation in a popup prompt. Once the order is sent to the user, they can confirm by clicking the Received button displayed on the page.

Error checking and edge cases

To eliminate faulty logic requests, such as the user ordering more items than are available in stock or two people checking out at the same time that total exceeds what is available in stock, I have added multiple checks and visible feedback for the users.

a) On the Product Details page, the decrement button is disabled when the quantity is at its default of 1. Similarly, the increment button is disabled when the amount of items in the cart and on the screen equals the stock quantity. The on-change event of the input field will also check for negative and decimal values and reset them to the default 1.
b) On the Cart page, the same logic is implemented, with the difference that clicking the decrement button when the item quantity is at one will trigger a popup to confirm if you are trying to remove the item from your cart.
c) On the Checkout page, the query will be refreshed to recheck the stock value before sending a request to the server and proceeding with order creation.

Key takeaways

Developing the order placement and monitoring features has deepened my comprehension of the e-commerce business workflow and robust error-checking practices. Utilizing OutSystems' Blocks for cart management ensures consistency and easy modifications, enhancing the user experience. The order creation process involves a seamless flow, from adjusting item quantities in the cart to selecting shipping options and placing orders, with backend data actions executing essential tasks. I have implemented multiple checks and visible feedback to address potential issues like stock inconsistencies and simultaneous checkouts. Decrement and increment buttons are intelligently disabled based on stock availability, and error prompts confirm user actions to prevent accidental removal. The Checkout page dynamically refreshes stock values before order creation, ensuring data accuracy. The Order Details page provides users a comprehensive view of their orders, allowing for cancellations before shipment confirmation.

5.4. Product Review and Rating

After an order is successfully delivered, users can leave reviews and ratings of their purchased products. The Reviews page divides the products into two tabs: Not Yet Reviewed and Reviewed, with Not Yet Reviewed having a button that redirects the site to a form where users can add images of the products on hand, add comments, and rate the products out of five. After submission, the product will be updated with new user rating points.

Key takeaways

Implementing the product review and rating feature on the Reviews page has enriched my skills in dynamic list management in OutSystems, facilitating organised displays of products in both the Not Yet Reviewed and Reviewed tabs. Beyond the immediate task, I gained insights into data modelling for future feature expansion, ensuring the system's adaptability to evolving requirements. Learning effective strategies for data migration was crucial, allowing seamless incorporation of new data and accommodating feature-specific needs without compromising system integrity. This experience underscores the significance of addressing current functionalities and preparing the system for scalable enhancements, showcasing a proactive approach to system development and maintenance in response to evolving user needs.

5.5. User Account Management

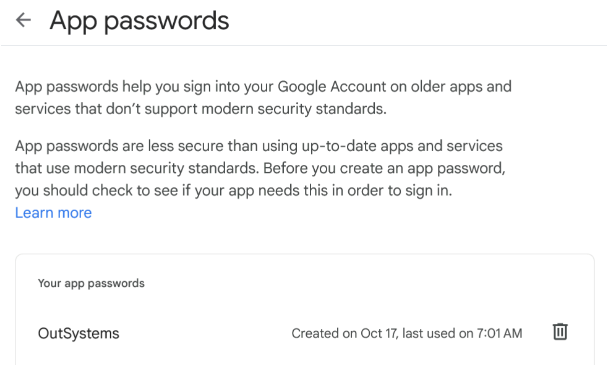
In the application, the user can update their account credentials. There are three tabs on the Account page: Basic Details, Password, and Address. In the Basic Details tab, the user can view and update information like profile picture, name, mobile phone, and birthday, all of which will be validated with the same rules as the Registration form. In the Password tab, the user can update their passwords. First, they must enter their old ones and re-confirm the new password (password strength rules stay the same with the registration process to ensure account security). Upon update, the user will log out of the current session, clearing session and client variables, and must log in again with their new password. Users can create, update, or delete their addresses in the Address tab. An address record will include the physical address, recipient name, and phone number.

Key takeaways

Implementing the User Account Management feature involved reinforcing security practices by applying stringent validation rules to basic details updates, ensuring data integrity. Integrating password strength rules during password updates enhances account security, and the strategic use of client variables and sessions in OutSystems facilitates secure logout and reauthentication upon password changes. Proficiency in managing user sessions ensures a seamless and protected account management experience. The dynamic address management feature allows users to create, update, or delete addresses, enhancing account personalisation. This task deepened my understanding of security practices in account management. It strengthened my skills in efficiently utilising client variables and sessions, contributing to a comprehensive and user-friendly account management process prioritising security and flexibility.

5.6. SMTP (Simple Mail Transfer Protocol)

An essential aspect of any business service, especially in e-commerce, is sending emails. Setting up an SMTP (Simple Mail Transfer Protocol) will assist the business by sending authentication tokens, notifications, and even order receipts. Using my account, I have used the Google Mail SMTP server for development and testing purposes. By setting up an app password and giving access credentials to the OutSystems administrative console, the environment can send emails to the user.



App passwords

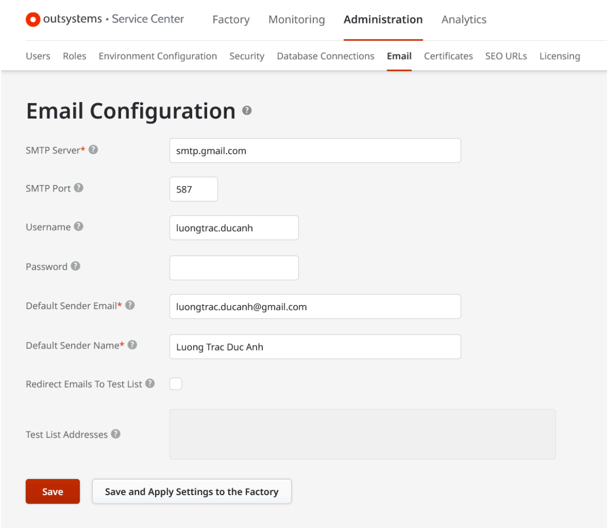
App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.
Learn more

Your app passwords

OutSystems Created on Oct 17, last used on 7:01 AM

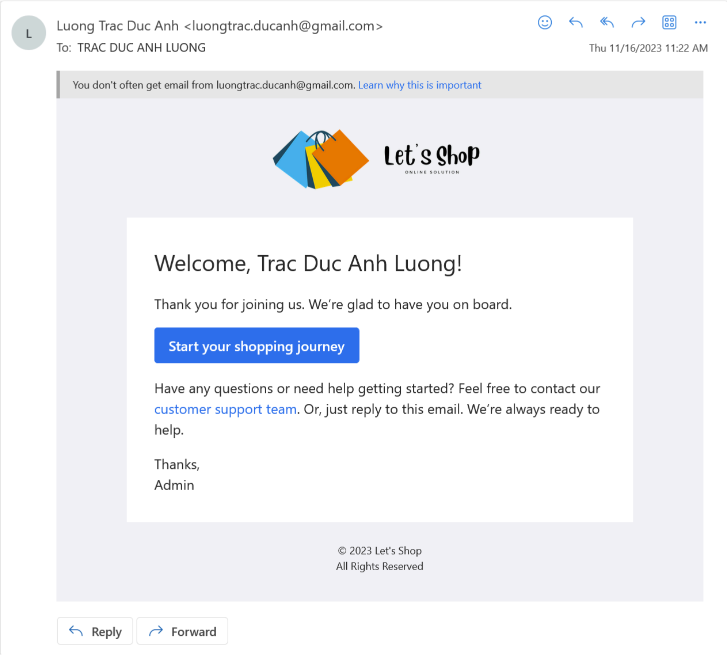
Figure 3: Setting up app password using Google Mail



Email Configuration

SMTP Server*: smtp.gmail.com
SMTP Port: 587
Username: luongtrac.ducanh
Password:
Default Sender Email*: luongtrac.ducanh@gmail.com
Default Sender Name*: Luong Trac Duc Anh
Redirect Emails To Test List: □
Test List Addresses:

Figure 4: Adding app password and set up credentials for SMTP server



Luong Trac Duc Anh <luongtrac.ducanh@gmail.com>
To: TRAC DUC ANH LUONG
Thu 11/16/2023 11:22 AM

You don't often get email from luongtrac.ducanh@gmail.com. Learn why this is important

Let's Shop
ONLINE SOLUTION

Welcome, Trac Duc Anh Luong!

Thank you for joining us. We're glad to have you on board.

Start your shopping journey

Have any questions or need help getting started? Feel free to contact our customer support team. Or, just reply to this email. We're always ready to help.

Thanks,
Admin

© 2023 Let's Shop
All Rights Reserved

Figure 5: Email sent upon success account registration

Key takeaways

Implementing SMTP using the Google Mail server in the OutSystems environment has streamlined crucial business communication, enabling the sending of authentication tokens, notifications, and order receipts. The process involves setting up an app password for secure authentication and configuring SMTP server credentials within the OutSystems administrative console. This integration ensures a reliable and efficient email communication channel. The hands-on experience with emails provided insights into email protocols, allowing for the creation of reusable email templates and styling using OutSystems' robust UI framework. This enhances the aesthetic appeal of emails and streamlines the development process. The practical application of email functionalities underscores the importance of secure and visually appealing communication in business services, mainly e-commerce. In summary, working with SMTP and emails in the OutSystems environment has improved communication channels and enhanced skills in configuring secure email protocols and creating visually appealing, reusable email templates.

5.7. Automation

OutSystems offers a unique automation feature named Timer. A Timer is an OutSystems tool that allows the execution of application logic periodically at a scheduled time [5]. These are also known as batch jobs. Different Timers can be executed simultaneously, but the same Timer never has more than one execution at a time. There are different use cases when it comes to Timers:

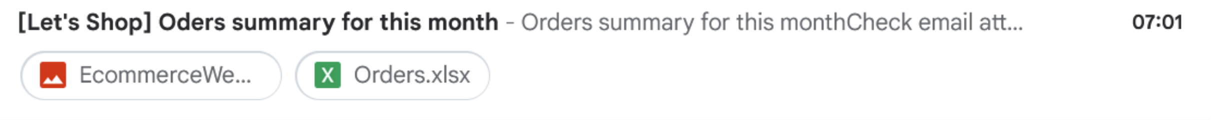
a) Scheduled Jobs: Execute the same job every day at the same time. For example, email subscribers every day at 4 am with digest news. Create a Timer to execute an Action that emails subscribers daily at 4 am.

b) Executing Long Running Actions: Execute application logic that usually takes a long time to finish. For example, the system must archive many database records at 2 am on the 1st day of every month. It takes about 2 hours. Create a Timer to execute an Action that archives records and set it to run at 2 am on 1st day of every month with the default timeout of 150 minutes (this value can be adjusted in the Service Center).

For my project, I have created two actions that related to sending emails:

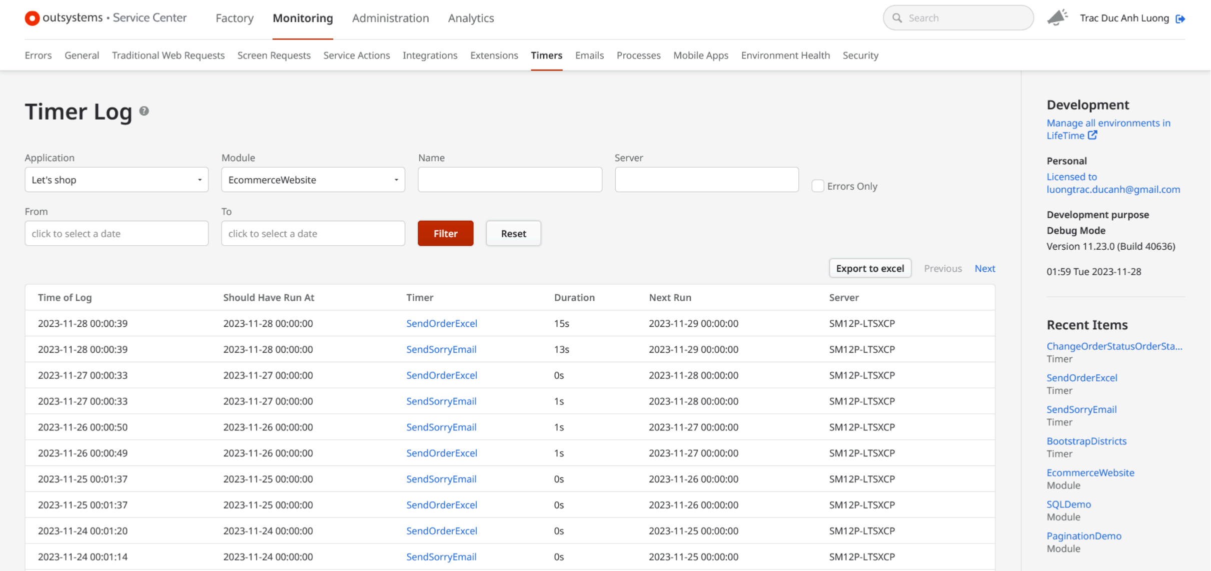
a) Every day at 12 am UTC or 7 am GMT, the `SendSorryEmail` action will run. It queries all orders that have the status `Shipped`. The store has a policy in terms of shipping time. Express will take four days, Rush will take two days, and Economical will take seven days. If the time from the order date to the time the action runs is longer than the planned days, the system will send a notification email to the user.

b) At the beginning of every month, the system will run the `SendOrderExcel` action and send the admin an Excel file summarising the orders placed during the previous month. The Excel file is created using a template and a Forge component to fill in the data. I will mention the Forge component in the Admin Flow (ManageOrder page).



[Let's Shop] Oders summary for this month - Orders summary for this monthCheck email att... 07:01

Figure 6: Order summary spreadsheet sent to the admin at 7 am GMT



Timer Log

Application: Let's shop
Module: EcommerceWebsite
Name:
Server:
□ Errors Only
From: click to select a date
To: click to select a date

Time of Log	Should Have Run At	Timer	Duration	Next Run	Server
2023-11-28 00:00:39	2023-11-28 00:00:00	SendOrderExcel	15s	2023-11-29 00:00:00	SM12P-LTSXCP
2023-11-28 00:00:39	2023-11-28 00:00:00	SendSorryEmail	13s	2023-11-29 00:00:00	SM12P-LTSXCP
2023-11-27 00:00:33	2023-11-27 00:00:00	SendOrderExcel	0s	2023-11-28 00:00:00	SM12P-LTSXCP
2023-11-27 00:00:33	2023-11-27 00:00:00	SendSorryEmail	1s	2023-11-28 00:00:00	SM12P-LTSXCP
2023-11-26 00:00:50	2023-11-26 00:00:00	SendSorryEmail	1s	2023-11-27 00:00:00	SM12P-LTSXCP
2023-11-26 00:00:49	2023-11-26 00:00:00	SendOrderExcel	1s	2023-11-27 00:00:00	SM12P-LTSXCP
2023-11-25 00:01:37	2023-11-25 00:00:00	SendSorryEmail	0s	2023-11-26 00:00:00	SM12P-LTSXCP
2023-11-25 00:01:37	2023-11-25 00:00:00	SendOrderExcel	0s	2023-11-26 00:00:00	SM12P-LTSXCP
2023-11-24 00:01:20	2023-11-24 00:00:00	SendOrderExcel	0s	2023-11-25 00:00:00	SM12P-LTSXCP
2023-11-24 00:01:14	2023-11-24 00:00:00	SendSorryEmail	0s	2023-11-25 00:00:00	SM12P-LTSXCP

Development
Manage all environments in LifeTime

Personal
Licensed to luongtrac.ducanh@gmail.com

Development purpose
Debug Mode
Version 11.23.0 (Build 40636)

01:59 Tue 2023-11-28

Recent Items
ChangeOrderStatusOrderSta... Timer
SendOrderExcel Timer
SendSorryEmail Timer
BootstrapDistricts Timer
EcommerceWebsite Module
SQLDemo Module
PaginationDemo Module

Figure 7: Timer Log and Monitoring

Key takeaways

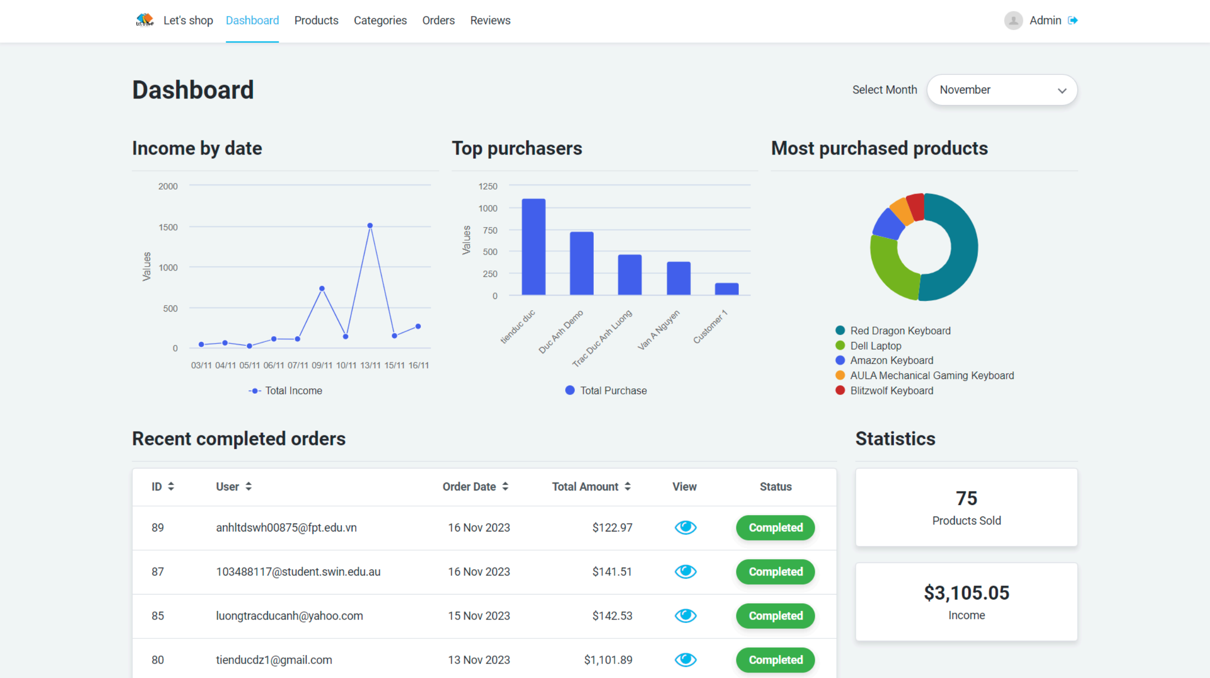
OutSystems Timer significantly streamlines the execution of scheduled jobs and manages long-running actions in the project, exemplified by tasks such as sending daily apology emails for shipping delays and generating monthly order summaries. The system's autonomy in handling these tasks at specified times ensures timely and efficient processes, minimising the need for manual intervention. The Timer's log and monitoring features contribute to a streamlined, error-resilient automation architecture, providing a clear view of executions and facilitating effective error handling. This implementation underscores the efficiency gained through leveraging Timers for routine tasks, including apology emails and order summaries, reducing manual effort and enhancing overall system reliability. In summary, the OutSystems Timer has automated everyday tasks seamlessly and offered insights into effective monitoring and error handling, highlighting its broader applications in business services and emphasising the pivotal role of systematic automation in maintaining a reliable and efficient system architecture [6].

6. Admin Flow

To visually represent the customer's interaction flow, I have created a UML diagram consisting of each screen's attributes and methods, which you can view in Figure 11 in the Appendices section. The flow consists of Admin Panels and their logic, following a master-detail architecture.

6.1. Dashboard and Data Analytics

Any business will generate data from their operation time [7]. In the context of my project, the e-commerce website will have data related to orders and user preferences after purchasing the store's products. Drawing insights from these data will help store owners realise trends and make critical choices. For this, I have created an Admin Dashboard (Figure 8) that conceptualises the data through graphs and table figures. Necessary information can be drawn from these figures, such as income by date, top purchasers, and most purchased products. Summary statistics from the total income and products sold will also be displayed in two live counters. Finally, there is a dropdown containing the months the admin can select, and the data will be refreshed to the chosen time accordingly.



ID	User	Order Date	Total Amount	View	Status
89	anhltdswh00875@fpt.edu.vn	16 Nov 2023	$122.97	👁	Completed
87	103488117@student.swin.edu.au	16 Nov 2023	$141.51	👁	Completed
85	luongtracducanh@yahoo.com	15 Nov 2023	$142.53	👁	Completed
80	tienducdz1@gmail.com	13 Nov 2023	$1,101.89	👁	Completed

75
Products Sold

$3,105.05
Income

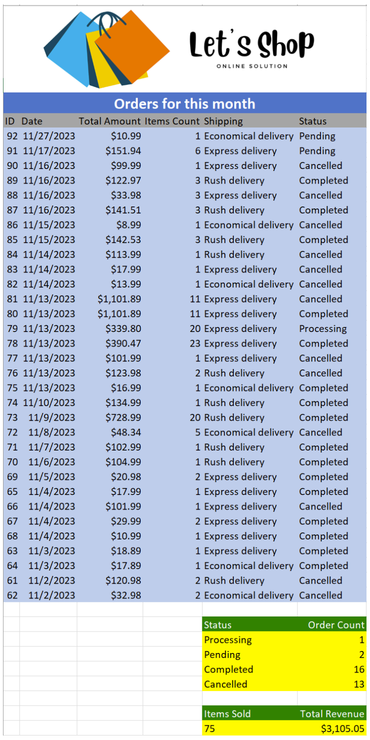
Figure 8: Admin Dashboard

Key takeaways

Creating the Admin Dashboard in my e-commerce project was a transformative experience using JavaScript graph libraries like D3.js and Chart.js to visualise critical business data. This dashboard is a powerful tool for store owners, offering instant insights into income trends, top purchasers, and popular products. Including live counters ensures real-time monitoring of critical metrics such as total income and products sold, eliminating the need for manual calculations [8]. The user-friendly dropdown for selecting specific months allows administrators to dynamically refresh data, facilitating customised analyses over different time frames. This project strengthened my data visualisation skills and underscored the importance of prioritising business-relevant insights for effective decision-making. In essence, the Admin Dashboard stands as a testament to the value of streamlined, real-time data interactions and strategic visualisation in enhancing the analytical capabilities of an e-commerce platform.

6.2. Order Management

In the Order Management screen, the admin can view the order count of each status, filter orders using the status tab, view the details of an order for packaging and delivery, and update the order status to notify the users. The admin can export an order summary, similar to the `SendOrderExcel` timer action mentioned in the Automation section. This action was implemented using a Forge component named *Advanced Excel*. Forge in OutSystems can be described as a market where developers write their libraries on top of the ASP.NET framework and publish them for other community members to incorporate into their applications. After extensive research and testing, I found the *Advanced Excel* Forge the most suitable for my project as it was packed with high-level Excel functionalities, well-documented, and famous in the OutSystems community. The Forge had high performance when used as a Timer action or during the export action on the Order Management page.



Orders for this month

ID	Date	Total Amount	Items Count	Shipping	Status
92	11/27/2023	$10.99	1	Economical delivery	Pending
91	11/17/2023	$151.94	6	Express delivery	Pending
90	11/16/2023	$99.99	1	Express delivery	Cancelled
89	11/16/2023	$122.97	3	Rush delivery	Completed
88	11/16/2023	$33.98	3	Express delivery	Cancelled
87	11/16/2023	$141.51	3	Rush delivery	Completed
86	11/15/2023	$8.99	1	Economical delivery	Cancelled
85	11/15/2023	$142.53	3	Rush delivery	Completed
84	11/14/2023	$113.99	1	Rush delivery	Cancelled
83	11/14/2023	$17.99	1	Express delivery	Cancelled
82	11/14/2023	$13.99	1	Economical delivery	Cancelled
81	11/13/2023	$1,101.89	11	Express delivery	Cancelled
80	11/13/2023	$1,101.89	11	Express delivery	Completed
79	11/13/2023	$339.80	20	Express delivery	Processing
78	11/13/2023	$390.47	23	Express delivery	Completed
77	11/13/2023	$101.99	1	Express delivery	Cancelled
76	11/13/2023	$123.98	2	Rush delivery	Cancelled
75	11/13/2023	$16.99	1	Economical delivery	Completed
74	11/10/2023	$134.99	1	Rush delivery	Completed
73	11/9/2023	$728.99	20	Rush delivery	Completed
72	11/8/2023	$48.34	5	Economical delivery	Cancelled
71	11/7/2023	$102.99	1	Rush delivery	Completed
70	11/6/2023	$104.99	1	Rush delivery	Completed
69	11/5/2023	$20.98	2	Express delivery	Completed
65	11/4/2023	$17.99	1	Express delivery	Completed
66	11/4/2023	$101.99	1	Express delivery	Cancelled
67	11/4/2023	$29.99	2	Express delivery	Completed
68	11/4/2023	$10.99	1	Express delivery	Completed
63	11/3/2023	$18.89	1	Express delivery	Completed
64	11/3/2023	$17.89	1	Economical delivery	Completed
61	11/2/2023	$120.98	2	Rush delivery	Cancelled
62	11/2/2023	$32.98	2	Economical delivery	Cancelled

Status	Order Count
Processing	1
Pending	2
Completed	16
Cancelled	13

Items Sold	Total Revenue
75	$3,105.05

Figure 9: Exported Excel Template

Key takeaways

The Order Management screen in my project facilitates efficient administration, allowing admins to view order counts by status, filter orders, inspect details for packaging and delivery, and update statuses to notify users. Integration of the *Advanced Excel* Forge component streamlines order summary exports, showcasing the benefits of leveraging community-contributed solutions. Chosen for its high-level Excel functionalities, documentation, and community popularity, this component enhances performance during timer actions and order export processes. The experience emphasizes the importance of user-friendly interfaces for order management and highlights the OutSystems community's collaborative nature by utilising Forge components. In addition, the backend logic development for Order Management exposes the intricacies of business processes, queue actions, and order processing workflows. This endeavour combines efficient order management, community collaboration, and insights into backend intricacies for a holistic understanding of business processes.

6.3. Product Management, Category Management, and Review Summary

The Product and Category sections are seamlessly created with the help of the OutSystems scaffolding function. Creating List and Detail screens is very fast, with scaffolding for Reactive Web Apps. By dragging an entity (data table) to any UI flow, two new screens completed with working actions and validations will be ready to publish. There are queries, list displays, pagination, sorting, and form reading with just one click. Corporating new functions was also more accessible than traditional coding with boilerplate code.

Key takeaways

OutSystems' scaffolding function proved instrumental in swiftly creating and managing Product and Category sections, offering a seamless development experience for Reactive Web Apps. The drag-and-drop capability facilitated the creation of fully functional screens with essential features like queries, list displays, pagination, sorting, and form reading, eliminating the need for labour-intensive coding. This experience underscored the efficiency of low-code practices, enabling rapid development without traditional boilerplate code. Learning to leverage OutSystems' fast and extensible functions emphasized the evolving landscape of low-code development, where features like scaffolding play a crucial role in simplifying and accelerating the development process. Eliminating boilerplate code streamlined development and contributed to cleaner, maintainable code, aligning with the contemporary trend towards more efficient and agile application development practices. The key takeaways encompass the power of OutSystems' low-code capabilities, the impact of emerging trends, and the tangible benefits of eliminating redundant coding efforts.

7. Final Project Presentation and Interview

At the end of my 8-week internship, I engaged in a final project presentation and interview with the company's esteemed team of product managers and developers. The session was successful, marked by positive feedback from my supervisors, who acknowledged the project's strengths while providing valuable suggestions for further improvement and development. The collaborative environment fostered insightful discussions, enhancing my understanding of OutSystems and the nuances of low-code development platforms. The technical interview, a reflection of my hands-on experience with an actual project, unfolded smoothly. The immersion in practical application significantly contributed to my confidence and proficiency in discussing technical aspects. As I document this report, I eagerly await a return offer from OCG, which I anticipate will materialize at the end of the week. This internship journey has enhanced my technical skills and provided a valuable platform for professional growth and potential long-term collaboration with the company.

8. Conclusion

Going through this 8-week internship with OCG Technology JSC has been a rewarding journey, immersing me in the dynamic realm of OutSystems web development and exposing me to the intricate workings of a technology company dedicated to providing innovative IT solutions. As a Software Developer Intern, the intensive training, mentorship, and hands-on project development have significantly contributed to my technical and professional growth. The personal project, focused on e-commerce, served as a practical playground to enhance my OutSystems skills, addressing many requirements and showcasing the platform's capabilities in creating responsive, performant, and user-friendly web applications.

The internship structure, encompassing training sessions, mentor guidance, and project deliverables, was a well-crafted blend that facilitated a comprehensive learning experience. The final project presentation and interview were instrumental in showcasing my achievements and eliciting constructive feedback, paving the way for continuous improvement. As I await a return offer, I reflect on the successes of fulfilling 19 out of 24 project requirements, demonstrating a solid grasp of OutSystems functionalities and high-level development concepts.

In conclusion, this internship has been invaluable in my academic and professional journey, equipping me with practical skills, industry insights, and a deeper understanding of low-code development. I express my gratitude to OCG Technology JSC for this enriching opportunity, and I look forward to potential future collaborations in the ever-evolving landscape of IT solutions.

9. Reflection

Engaging in an 8-week internship with OCG Technology JSC provided a profound exploration of OutSystems web development and a rich intercultural experience within the context of a Vietnamese technology company. This reflection will delve into the intercultural aspects of the internship, exploring how cultural nuances influenced communication, teamwork, and personal development. Additionally, it will draw on relevant academic literature to provide theoretical insights into the intercultural dynamics encountered during the internship.

9.1. Intercultural Communication and Collaboration

Schnurr's [9] exploration of professional communication serves as a valuable lens to analyze the intricacies of intercultural communication within the workplace. The diverse team at OCG, comprising individuals from various cultural backgrounds, highlighted the need for a deep understanding of different communication styles and preferences. Zhu's [10] situated genre approach becomes particularly relevant, emphasizing the importance of adapting communication strategies in cross-cultural settings. The collaborative development environment at OCG fostered a rich exchange of perspectives, enriching problem-solving approaches and cultivating a dynamic and innovative atmosphere.

9.2. Cultural Adaptation and the Learning Curve

Ward, Bochner, and Furnham's [11] insights into culture shock shed light on the initial challenges faced when adapting to a new work environment. The learning curve at OCG involved not only comprehending the technical aspects of the OutSystems platform but also navigating the cultural nuances embedded in daily interactions. As outlined by Ward et al., the psychological impact of cultural adaptation manifested as a process of continuous adjustment, eventually leading to a more comfortable and effective integration into the workplace culture.

9.3. Conflict Resolution and Team Dynamics

Levine's [12] strategies for turning conflict into collaboration became instrumental in navigating moments of divergence within the team. As conflicts inevitably arose due to differences in approach or communication styles, employing resolution techniques became crucial. The reflective analysis, drawing on Dwyer and Hopwood [13], emphasizes the role of effective business communication in mitigating conflicts and fostering a collaborative environment.

9.4. Skills Development and Employability

Gill's [14] examination of graduate employability skills through internships aligns seamlessly with the experiential aspect of the internship at OCG. Acquiring technical expertise in OutSystems was paralleled by developing crucial employability skills, including communication, problem-solving, and adaptability. The practical application of theoretical knowledge, as Gill advocates, underscored the relevance of a holistic skill set in the professional realm.

9.5. Reflective Practice

Applying the Gibbs Reflective Cycle [15] provides a structured framework for analyzing the internship experience. The cycle comprehensively explores the event, feelings, evaluation, analysis, conclusion, and action plan. Reflecting on Dwyer and Hopwood's intercultural dimensions and insights, it becomes evident that acknowledging cultural differences, fostering open communication, and embracing a learning mindset were pivotal elements in navigating the challenges and capitalizing on the opportunities presented during the internship.

9.6. Conclusion and Future Development

In conclusion, the intercultural experience gained during the internship at OCG illuminated the significance of cultural intelligence (CQ) in a globalized workplace. The theoretical frameworks of Schnurr, Ward et al., Levine, Zhu, and Gill provided a deeper understanding of the interplay between cultural dynamics and professional development. Integrating these learnings into future professional endeavours will ensure a continued commitment to cultural awareness, effective communication, and collaborative success in diverse work environments. As the world becomes increasingly interconnected, CQ will undoubtedly play a more crucial role in fostering harmonious and productive cross-cultural collaborations.

References

[1] A. C. Society, “Web Developer ANZSCO 261212 | ACS skill assessment | ACSRPLReport.com”. Accessed: Nov. 27, 2023. [Online]. Available: https://www.acsrplreport.com/web-developer-anzsco-261212/

[2] “Database Design Fundamentals”, in *Database Principles and Technologies -- Based on Huawei GaussDB*, Singapore: Springer Nature Singapore, 2023, pp. 245–285. doi: 10.1007/978-981-19-3032-4_7.

[3] Q. Ji, “Study on Information Security Issues of E-Commerce”, *IOP Conference Series: Materials Science and Engineering*, p. 32050, 2018, doi: 10.1088/1757-899x/452/3/032050.

[4] R. Roth, “User Interface and User Experience (UI/UX) Design”, *Geographic Information Science & Technology Body of Knowledge*, 2017, doi: 10.22224/gistbok/2017.2.5.

[5] OutSystems, “Timer - OutSystems 11 Documentation”. Accessed: Nov. 28, 2023. [Online]. Available: https://success.outsystems.com/documentation/11/reference/outsystems_language/processes/timer/

[6] A. Astoray and L. Andrade-Arenas, “Implementation of an e-Commerce System for the Automation and Improvement of Commercial Management at a Business Level”, *International Journal of Advanced Computer Science and Applications*, 2021, doi: 10.14569/IJACSA.2021.0120177.

[7] B. Qiang Swan, “International Journal of Data Analytics”, *International journal of data analytics*, 2019, doi: 10.4018/ijda.

[8] U. Murthy, “An Analysis of the Effects of Continuous Monitoring Controls on E-Commerce System Performance”, *Journal of Information Systems*, pp. 29–47, 2004, doi: 10.2308/jis.2004.18.2.29.

[9] S. Schnurr, *Exploring Professional Communication. Language in Action*. 2012. doi: 10.4324/9780203095324.

[10] Y. Zhu, *The Routledge Handbook of Language and Professional Communication*. Routledge, 2014. doi: 10.4324/9781315851686.

[11] C. Ward, S. Bochner, and A. Furnham, *The Psychology of Culture Shock*. 2001. doi: 10.4324/9781003070696.

[12] S. Levine, *Getting to resolution: turning conflict into collaboration*. Berrett-Koehler Publishers, 1998.

[13] J. Dwyer and N. Hopwood, *The Business Communication Handbook*. Cengage Australia, 2015.

[14] R. J. Gill, “Graduate employability skills through online internships and projects during the COVID-19 Pandemic: An Australian example.”, *Journal of Teaching and Learning for Graduate Employability*, pp. 146–158, 2020, doi: 10.21153/jtlge2020vol11no1art946.

[15] M. Tools, “Gibbs Reflective Cycle”. Accessed: Nov. 28, 2023. [Online]. Available: https://www.mindtools.com/ano9qiu/gibbs-reflective-cycle

Appendices

This section will contain the report's figures of the application's UML diagrams.
URL for demo: https://personal-1gh7uatw.outsystemscloud.com/EcommerceWebsite/

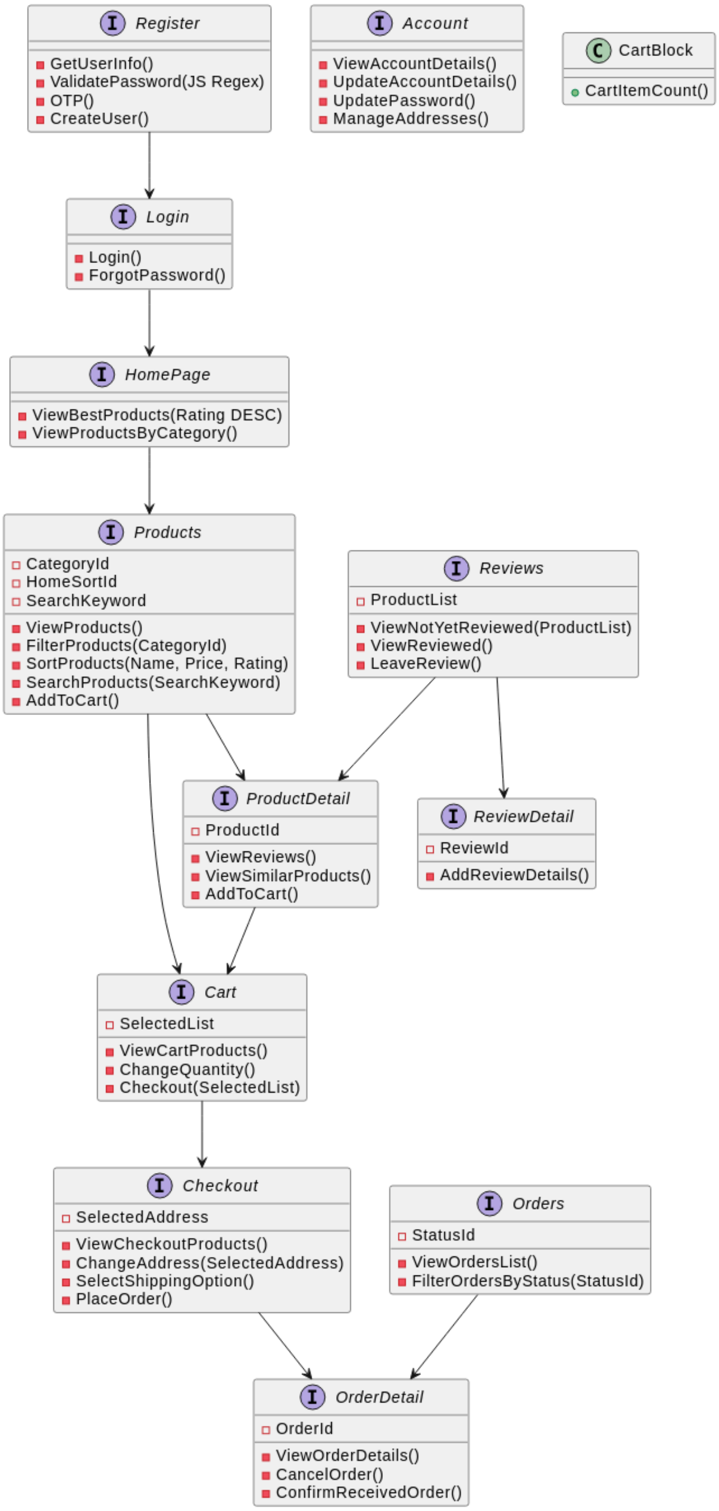


Figure 10: User Flow

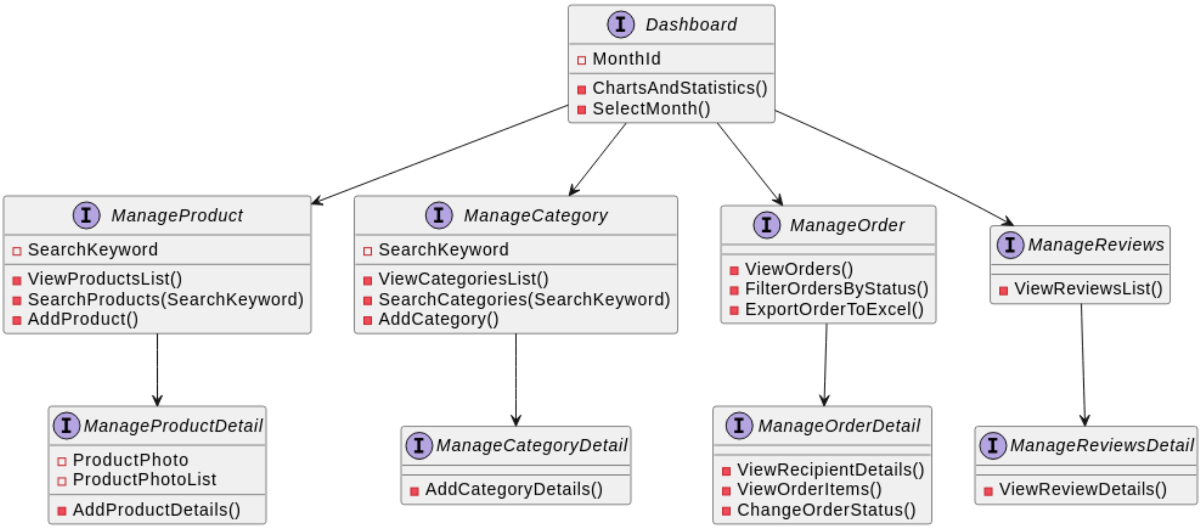


Figure 11: Admin Flow