

Grapes: NLP Web Craft

Design Document



Session: 2021-25

Submitted by:

Naqeeb Ahmed	2021-CS-616
Munib Ur Rehman	2021-CS-605
Ahsan Bashir	2021-CS-628

Supervised by:

Dr Zeeshan Ramzan

Department of Computer Science, New Campus **University of Engineering and Technology** Lahore, Pakistan

Table of Contents

List of Figures	3
1. Software Requirement Specification	4
1.1 Functional Requirement	4
1.2 Non-functional Requirement	5
2.Design specification	7
2.1 System Behavioral Design	7
2.1.1 Use case Diagram	7
2.1.2 Activity Diagram	17
2.1.3 State Diagram.....	29
2.1.4 Sequence Diagram	40
2.1.5 Collaboration Diagram	74
2.2 System Structure Design	110
2.2.1 Class Diagram	110
2.2.2 Component Diagram	111
2.2.3 Deployment Diagram	112
2.3 User Interface Design.....	113
2.3.1 Wireframes	113
2.4 Database Design.....	113
2.4.1 ER Diagram	113

List of Figures

Figure 1: Use Case Diagram	7
Figure 2: Sign up.....	17
Figure 3: Sign In.....	18
Figure 4: Manage Profile	19
Figure 5: Edit User.....	20
Figure 6: Manage Template	21
Figure 7: View Prompt History	22
Figure 8: Sign Up	23
Figure 9: Sign In.....	24
Figure 10: Create Website	25
Figure 11: GrapeJs Editing.....	26
Figure 12: View Features	27
Figure 13: Contact Us.....	28
Figure 14: Sign Up	29
Figure 15: Sign In.....	30
Figure 16: Manage Profile.....	31
Figure 17: manage Template	32
Figure 18: View Prompt History	33
Figure 19: Sign Up	34
Figure 20: Sign In.....	35
Figure 21: Create Website	36
Figure 22: GrapeJs Editing.....	37
Figure 23: View Features	38
Figure 24: Contact Us.....	39
Figure 25: Admin End.....	40
Figure 26: User End.....	44
Figure 27: Admin End Collaboration	74
Figure 28: User End Collaboration	78
Figure 29: Class Diagram	110
Figure 30: Component Diagram.....	111
Figure 31: Deployment Diagram	112
Figure 32: ER Diagram.....	113

Chapter 1

1. Software Requirement Specification

1.1 Functional Requirement

In developing a robust NLP Web Craft system, a comprehensive set of functional requirements has been defined. The functional requirements are grouped into four categories: Business requirements, administrative requirements, user requirements, and service requirements. Each category is assigned a unique identifier, such as B for business, A for administrative, U for user, and S for service requirements. The requirements are then numbered within each category, starting with 01.

The following list describes the main functions the system will perform:

- FR-B-01: The system shall provide web development business services aimed at individuals and small businesses.
- FR-B-02: The system shall target businesses looking for quick and efficient website solutions.
- FR-B-03: The system shall cater to content creators seeking to establish their online presence through customizable websites.
- FR-B-04: The system shall provide value-added services to enhance user experience, such as SEO optimization.
- FR-B-05: The system shall create business opportunities through affiliate marketing and partnerships with template providers.
- FR-B-06: The system shall offer subscription-based services for users seeking premium templates and advanced features.
- FR-A-01: The system shall enable and disable admin access for managing user accounts and roles.
- FR-A-02: The system shall allow admins to manage user roles and permissions, ensuring proper access control for different services.

- FR-A-03: The system shall share login IDs and new passwords with admins as needed for account recovery.
- FR-A-04: The system shall enable admins to view user activity logs to monitor interactions with the chatbot and website creation services.
- FR-U-01: The system shall provide a chatbot that interacts with users, asking questions about the type of website they want (e-commerce, blog, portfolio) to facilitate web development services.
- FR-U-02: The chatbot shall understand the user's responses and create a website plan based on their input, guiding them in content management and design services.
- FR-U-03: The system shall offer ready-to-use templates for different types of websites, including e-commerce, blogs, and portfolios, enhancing the template design services.
- FR-U-04: Users shall have the ability to customize design elements such as headers, footers, and color schemes for their websites, utilizing user experience (UX) services.
- FR-U-05: After editing, users shall be able to download their websites as a ZIP file containing all the HTML, CSS, and JS files, providing interoperability and export functionality.
- FR-U-06: The downloaded website files shall be fully editable by users for future modifications, ensuring ongoing user support and technical assistance.
- FR-S-01: The system shall provide a chatbot service to gather user requirements and preferences for website creation.
- FR-S-02: The system shall offer template design services, allowing users to select from various pre-built website templates.
- FR-S-03: The system shall facilitate customization services, enabling users to modify design elements of their websites.
- FR-S-04: The system shall support content management services, allowing users to organize and manage their website content effectively.
- FR-S-05: The system shall enable downloading services for users to export their website files as editable ZIP files.
- FR-S-06: The system shall provide ongoing technical support services for users to assist with any issues related to their websites

1.2 Non-functional Requirement

In developing the NLP Web Craft system, we aim to create a robust and user-friendly platform that empowers users to effortlessly build and customize their websites using advanced natural language

processing technologies. The project is designed to simplify the web development process for individuals and businesses, making it accessible even to those without technical expertise. By integrating a chatbot interface and ready-to-use templates, users can quickly generate website plans and customize design elements, ultimately enhancing their online presence.

To ensure the system meets the needs of diverse users, we've defined a comprehensive set of non-functional requirements. These requirements focus on various aspects of the system's performance,

usability, reliability, and scalability. Each requirement has been categorized and assigned a unique identifier for easy reference.

1. NFR-U-01: The interface must be simple and intuitive for non-technical users.
2. NFR-U-02: Users must be able to preview their websites in real-time while editing.
3. NFR-U-03: The system must be mobile-friendly for easy use across all devices.
4. NFR-U-04: Users must have the ability to undo/redo changes during website editing.
5. NFR-R-01: The system must handle temporary internet disruptions gracefully.
6. NFR-R-02: The system must gradually degrade without failure under heavy load.
7. NFR-S-01: The system must accommodate increases in data storage without performance degradation.
8. NFR-S-02: The system must support new features without requiring significant rewrites.
9. NFR-S-03: Specific components of the system must scale independently (e.g., the chatbot or API).
10. NFR-S-04: New templates and design elements must be easy to add over time.
11. NFR-P-01: The system must respond to user requests within 2 seconds.
12. NFR-P-02: API data searches must complete in 1 second or less.
13. NFR-P-03: Templates and previews must be updated in real-time during editing.
14. NFR-P-04: The system must function smoothly on low-bandwidth connections.

Chapter 2

2.Design specification

2.1 System Behavioral Design

2.1.1 Use case Diagram

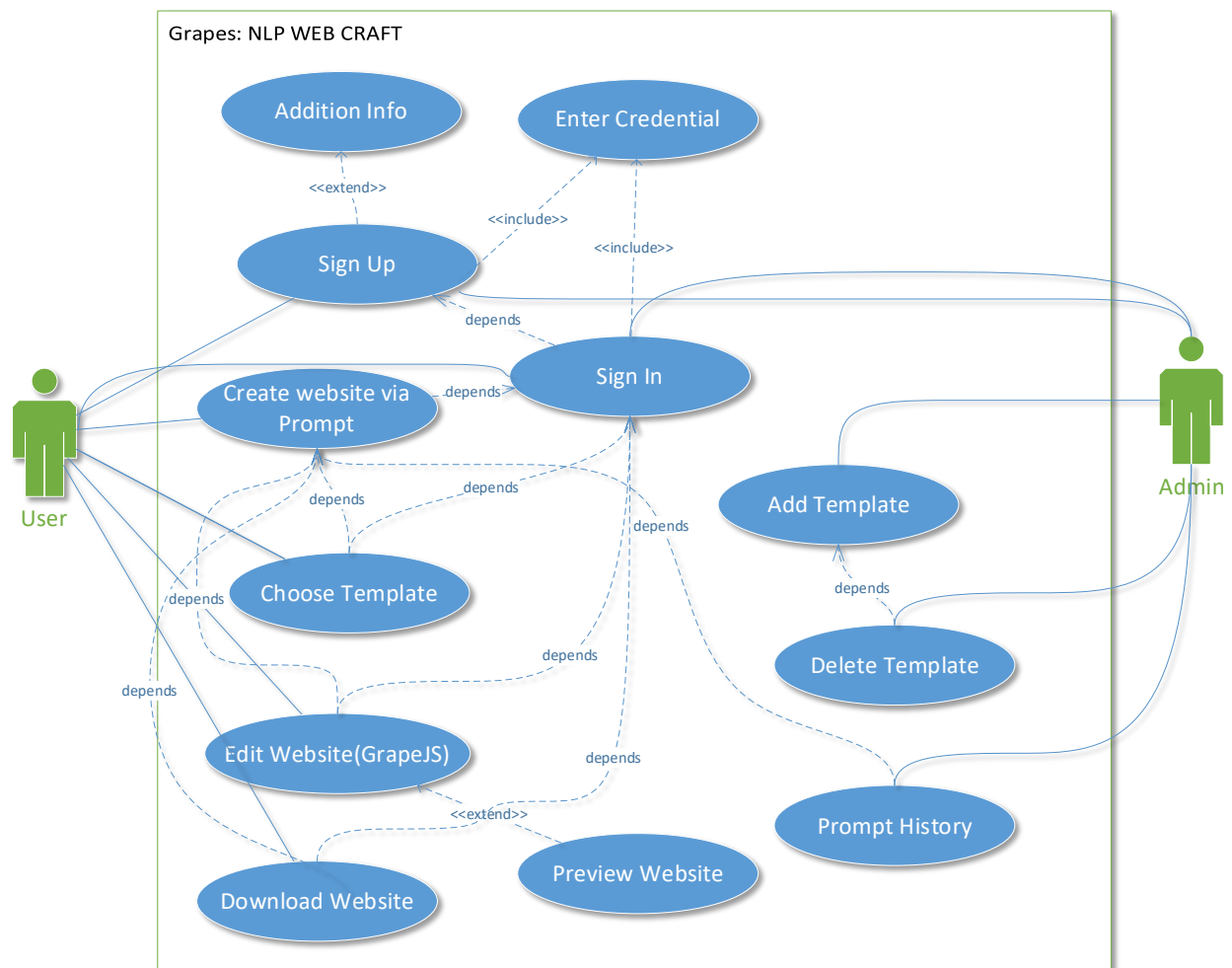


Figure 1: Use Case Diagram

Use Case Name: User Sign-Up for NLP Web Craft

Primary Actor: User

Trigger: When the "Create Account" button is pressed to register a new user.

Goal: Successfully create a user account to use the chatbot and website-building tools.

Precondition: The user is not already registered in the system.

Postcondition: User successfully creates an account and gains access to the system features.

Basic Flow:

1. User opens the NLP Web Craft platform.
2. User navigates to the "Sign-Up" section.
3. System prompts for user details (name, email, password).
4. User enters the required information.
5. System validates the information (e.g., email, password strength).
6. If valid, the system creates a new user account.

Alternate Flow:

N/A

Quality Requirements:

1. The sign-up process should be secure (password encryption).
2. The system should provide a better user experience.

Exception: If there is a technical issue (e.g., server/database failure), the system should display an error message and suggest retrying or contacting support.

Use Case Name: Admin Sign-Up

Primary Actor: Car Showroom Administrator

Trigger: When the "Sign Up" button is pressed to create a new admin account.

Goal: Successfully create a new admin account for accessing and managing the system.

Precondition: The administrator is not already registered in the system.

Postcondition: The administrator successfully creates an admin account and gains access to the administrative features of the application.

Basic Flow:

1. The administrator opens the car showroom application.
2. The administrator navigates to the "Admin Sign-Up" section.
3. The system prompts the administrator to provide necessary information (username, email, password, and contact details).
4. The administrator enters the required information.
5. The system validates the entered information (e.g., checks for valid email format, password strength).
6. If the information is valid, the system creates a new admin account.

Alternate Flow:

N/A

Quality Requirements:

1. The admin sign-up process should be secure, using encryption for passwords.
2. The system should handle the creation of admin accounts efficiently and securely.

Exception: If there are technical issues during the admin sign-up process (e.g., server error, database connection failure), the system should display a friendly error message and provide guidance on what the administrator should do next.

Use Case Name: User Sign-Up for NLP Web Craft

Primary Actor: User (Website Builder, Admin)

Trigger: When the "Log In" button is pressed to access the User dashboard.

Goal: Successfully create a user account to use the chatbot and website-building tools.

Precondition: The user is not already registered in the system.

Postcondition: User successfully creates an account and gains access to the system features.

Basic Flow:

1. User opens the NLP Web Craft platform.
2. User navigates to the "Sign-Up" section.
3. System prompts for user details (name, email, password).

4. User enters the required information.
5. System validates the information (e.g., email, password strength).
6. If valid, the system creates a new user account.

Alternate Flow:

N/A

Quality Requirements:

1. The sign-up process should be secure (password encryption).
2. The system should provide a seamless user experience.
3. Adequate logging should be implemented to track account creation for audit purposes.

Exception: If there is a technical issue (e.g., server/database failure), the system should display an error message and suggest retrying or contacting support.

Admin Sign-In

Use Case Name: Admin Sign-In

Primary Actor: Admin

Trigger: When the "Log In" button is pressed to access the admin dashboard.

Goal: Successfully log in to the admin account to manage the application.

Precondition: The administrator has a registered admin account.

Postcondition: The administrator is logged into the system and can access administrative features.

Basic Flow:

1. The administrator opens the application.
2. The administrator navigates to the "Admin Sign-In" section.
3. The system prompts the administrator to enter their username and password.
4. The administrator enters the required login credentials.
5. The system validates the credentials against the database.
6. If the credentials are valid, the system grants access and redirects the administrator to the admin dashboard.
7. The administrator can now manage the application.

Alternate Flow: N/A

Quality Requirements:

1. The sign-in process should be secure and protect against unauthorized access.
2. The system should provide feedback for failed login attempts.

Exception: If there are technical issues during the sign-in process (e.g., server error, database connection failure), the system should display a friendly error message and provide guidance on what the administrator should do next.

Create Website Via Prompt

Use Case Name: Create Website Via Prompt

Primary Actor: User

Trigger: When the "Create Website" button is pressed after selecting a type and writing a prompt.

Goal: Successfully create a website based on the user's prompt and selected type.

Precondition: The user is logged in to the system.

Postcondition: The website is created based on the user's specifications and is accessible for further editing.

Basic Flow:

1. The user opens the NLP Web Craft platform.
2. The user navigates to the "Create Website" section.
3. The system presents three options for website types: E-commerce, Blog, and Portfolio.
4. The user selects one of the options.
5. The system prompts the user to enter a description or prompt for the website.
6. The user writes and submits the prompt.

Alternate Flow:

N/A

Quality Requirements:

1. The website creation process should be efficient and responsive.
2. The system should handle user prompts securely and protect against malicious input.

Exception: If there are technical issues during the website creation process (e.g., server error, prompt processing failure), the system should display a friendly error message and provide guidance on what the user should do next.

Choose Template

Use Case Name: Choose Template

Primary Actor: User (Website Builder)

Trigger: When the "**Select Template**" button is pressed to choose a template for the website.

Goal: Successfully choose a template that fits the user's website design preferences.

Precondition: The user has created a website via prompt.

Postcondition: The selected template is applied to the website for further customization.

Basic Flow:

1. The user navigates to the "Choose Template" section.
2. The system displays a selection of available templates for E-commerce, Blog, and Portfolio websites.
3. The user selects a template that best suits their best.
4. The system applies the selected template to the website.
5. The user can now proceed to customize the website further with the chosen template.

Alternate Flow:

N/A

Quality Requirements:

1. The template selection process should be user-friendly and visually appealing.
2. The system should load templates efficiently to enhance user experience.
3. Adequate logging should be implemented to track template selections for auditing purposes.

Exception: If there are technical issues during the template selection process (e.g., server error, database failure), the system should display a friendly error message and provide guidance on what the user should do next.

Edit Website (GrapeJS)

Use Case Name: Edit Website Using GrapeJS

Primary Actor: User (Website Builder)

Trigger: When the "Edit" button is pressed to open the GrapeJS editor for customization.

Goal: Successfully modify and customize the website using GrapeJS editor.

Precondition: The user has selected a template for their website.

Postcondition: The changes made to the website are saved, and the user can preview the updated website.

Basic Flow:

1. The user navigates to the "Edit Website" section using GrapeJS.
2. The system loads the selected template into the GrapeJS editor.
3. The user interacts with the GrapeJS interface to modify elements (text, images, layout).
4. The user can add, remove, or rearrange components within the website.
5. The user previews the changes in real-time using the GrapeJS preview feature.
6. The user saves the changes once satisfied with the edits.
7. The system confirms that the changes have been successfully saved.

Alternate Flow:

N/A

Quality Requirements:

1. The GrapeJS editor should be intuitive and responsive, providing a seamless editing experience.
2. The system should ensure all changes are saved securely and efficiently.

Exception: If there are technical issues during the editing process (e.g., server error, failure to save changes), the system should display a friendly error message and provide guidance on what the user should do next.

Download Website

Use Case Name: Download Website

Primary Actor: User (Website Builder)

Trigger: When the "Download" button is pressed to download the completed website files.

Goal: Successfully download the website files in a zip format.

Precondition: The user has edited and finalized their website using the GrapeJS editor.

Postcondition: The website files are downloaded to the user's device.

Basic Flow:

1. The user navigates to the "Download Website" section.
2. The system presents an option to download the completed website.
3. The user clicks the "Download" button.
4. The system compiles the website files (HTML, CSS, JavaScript, assets) into a zip file.
5. The system prompts the user to save the zip file to their device.
6. The system confirms that the download is complete.

Alternate Flow:

N/A

Quality Requirements:

1. The download process should be efficient and secure.
2. The system should ensure that all necessary files are included in the zip package.

Exception: If there are technical issues during the download process (e.g., server error, failure to compile files), the system should display a friendly error message and provide guidance on what the user should do next.

Add Template

Use Case Name: Add Template

Primary Actor: Admin

Trigger: When the "Add Template" button is pressed to upload a new template.

Goal: Successfully add a new template to the system for user selection.

Precondition: The admin is logged into the system with appropriate privileges.

Postcondition: The new template is available for users to select when creating their websites.

Basic Flow:

1. The admin opens the NLP Web Craft platform.

2. The admin navigates to the "Add Template" section.
3. The system prompts the admin to upload the template files (HTML, CSS, JavaScript, assets).
4. The admin uploads the necessary files and enters relevant details (template name, description, category).
5. The system validates the uploaded files and details.

Alternate Flow:

N/A

Quality Requirements:

1. The template addition process should be secure and validate file types to prevent malicious uploads.
2. The system should handle template uploads efficiently and provide feedback on upload status.
3. Adequate logging should be implemented to track template additions for audit purposes.

Exception: If there are technical issues during the template addition process (e.g., server error, database failure), the system should display a friendly error message and provide guidance on what the admin should do next.

Delete Template

Use Case Name: Delete Template

Primary Actor: Admin

Trigger: When the "**Delete**" button is pressed to remove an existing template from the system.

Goal: Successfully delete a template from the system.

Precondition: The admin is logged into the system with appropriate privileges.

Postcondition: The selected template is removed from the system and is no longer available for user selection.

Basic Flow:

1. The admin navigates to the "Manage Templates" section.
2. The system displays a list of existing templates.
3. The admin selects the template they wish to delete.

4. The system prompts the admin to confirm the deletion.
5. The system confirms that the template has been successfully deleted.

Alternate Flow:

N/A

Quality Requirements:

1. The deletion process should be secure, preventing unauthorized access to template management features.

Exception: If there are technical issues during the deletion process (e.g., server error, database failure), the system should display a friendly error message and provide guidance on what the admin should do next.

Prompt History for Admin

Use Case Name: View Prompt History

Primary Actor: Admin (Car Showroom Administrator)

Trigger: When the "**View Prompt History**" button is pressed to access the list of previous prompts.

Goal: Successfully access and review the prompt history to monitor user interactions.

Precondition: The admin is logged into the system with appropriate privileges.

Postcondition: The admin can view the details of previous prompts submitted by users.

Basic Flow:

1. The admin opens the NLP Web Craft platform.
2. The admin navigates to the "Prompt History" section.
3. The system retrieves and displays a list of previously submitted prompts, including details such as user ID, prompt content, timestamp, and status.

Alternate Flow:

N/A

Quality Requirements:

1. The prompt history retrieval process should be efficient and fast to enhance the admin's ability to monitor user interactions.
2. The system should ensure that all user data is displayed securely and without exposing sensitive information.

Exception: If there are technical issues during the retrieval of prompt history (e.g., server error, database failure), the system should display a friendly error message and provide guidance on what the admin should do next.

2.1.2 Activity Diagram

2.1.2.1 Admin End

Sign Up:

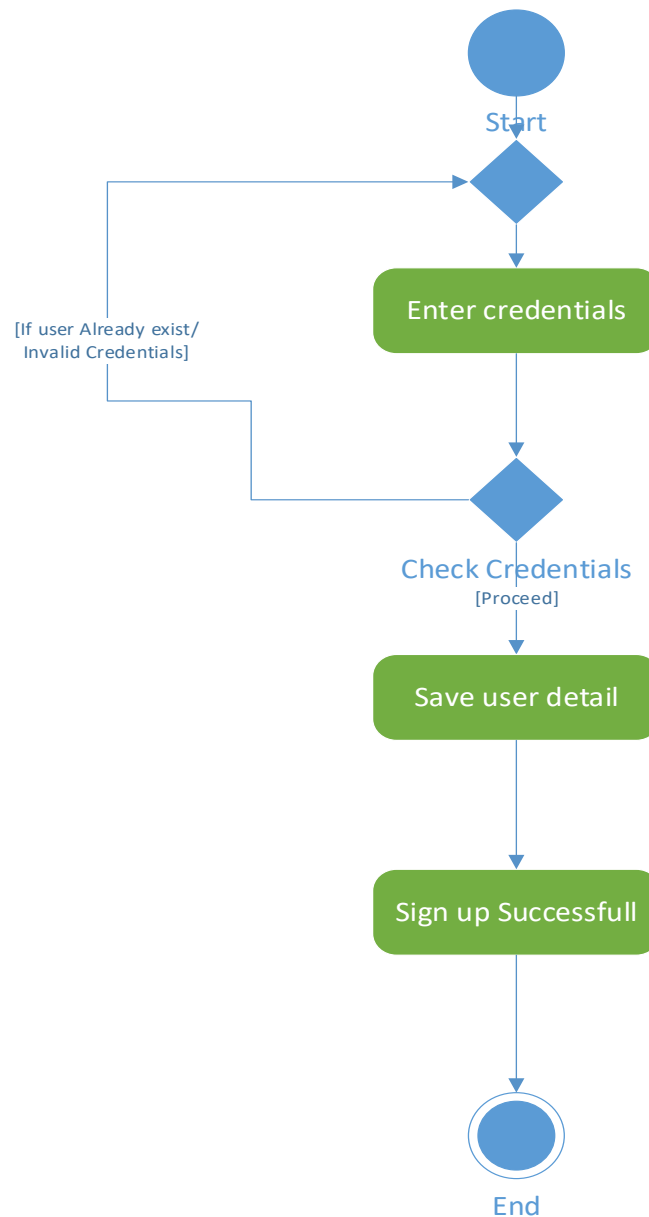


Figure 2: Sign up

Sign In:

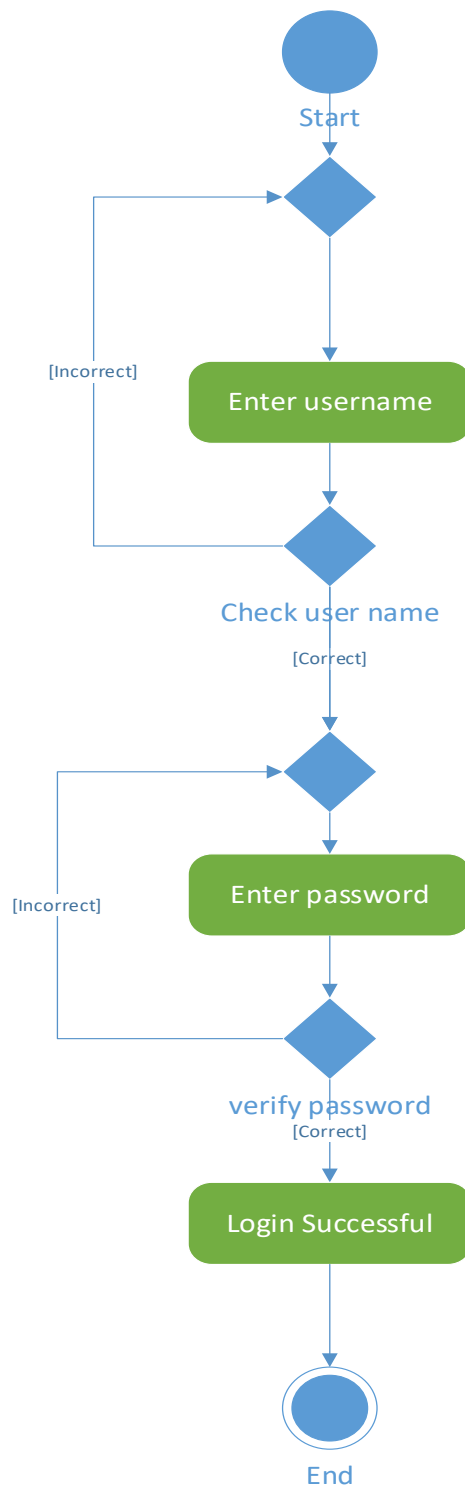


Figure 3: Sign In

Manage Profile

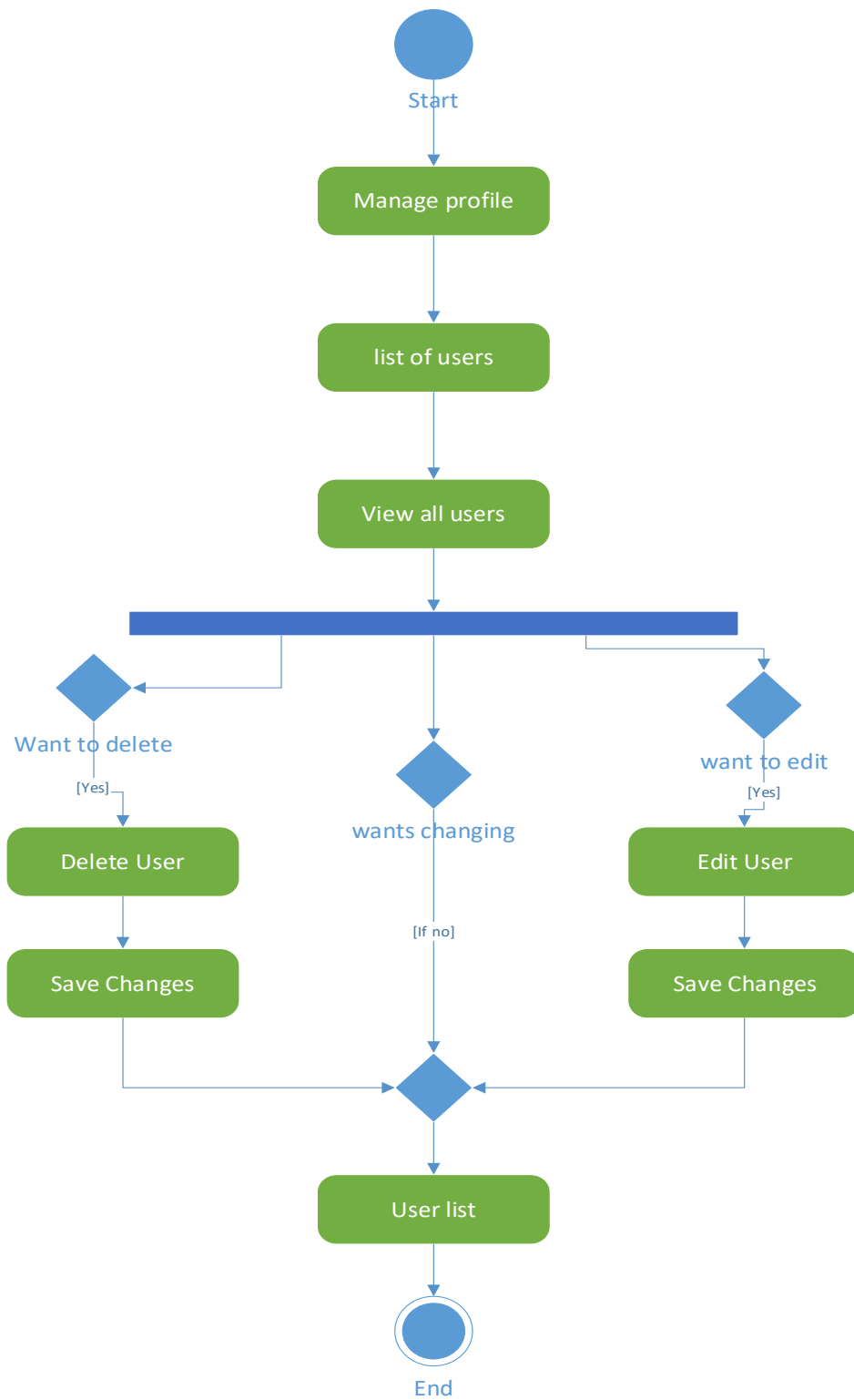


Figure 4: Manage Profile

Edit user:

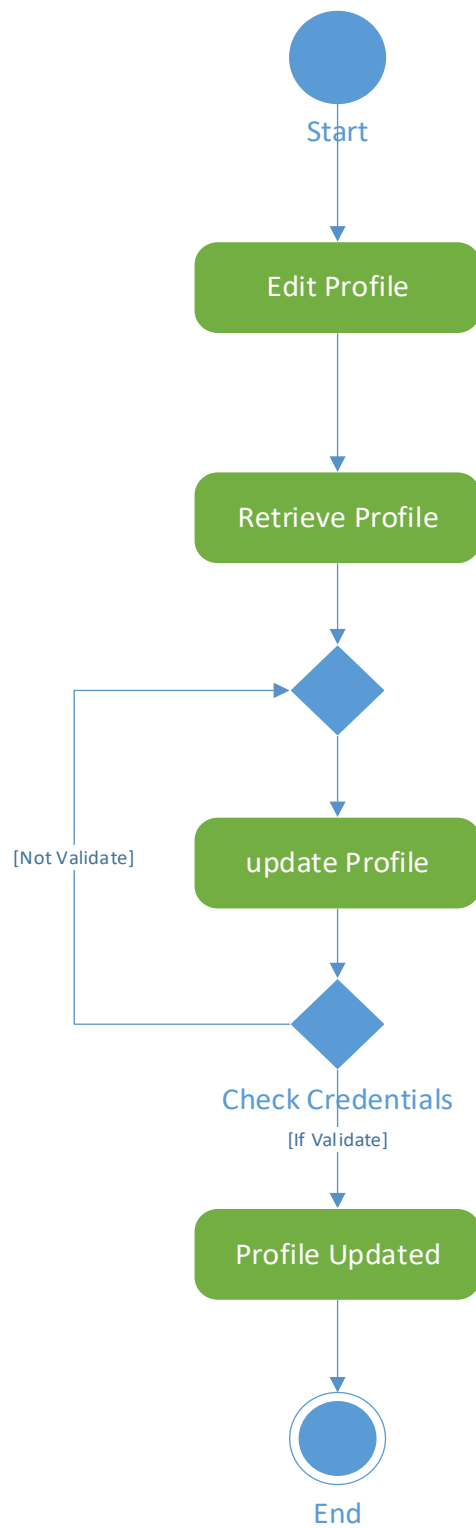


Figure 5: Edit User

Manage Templates:

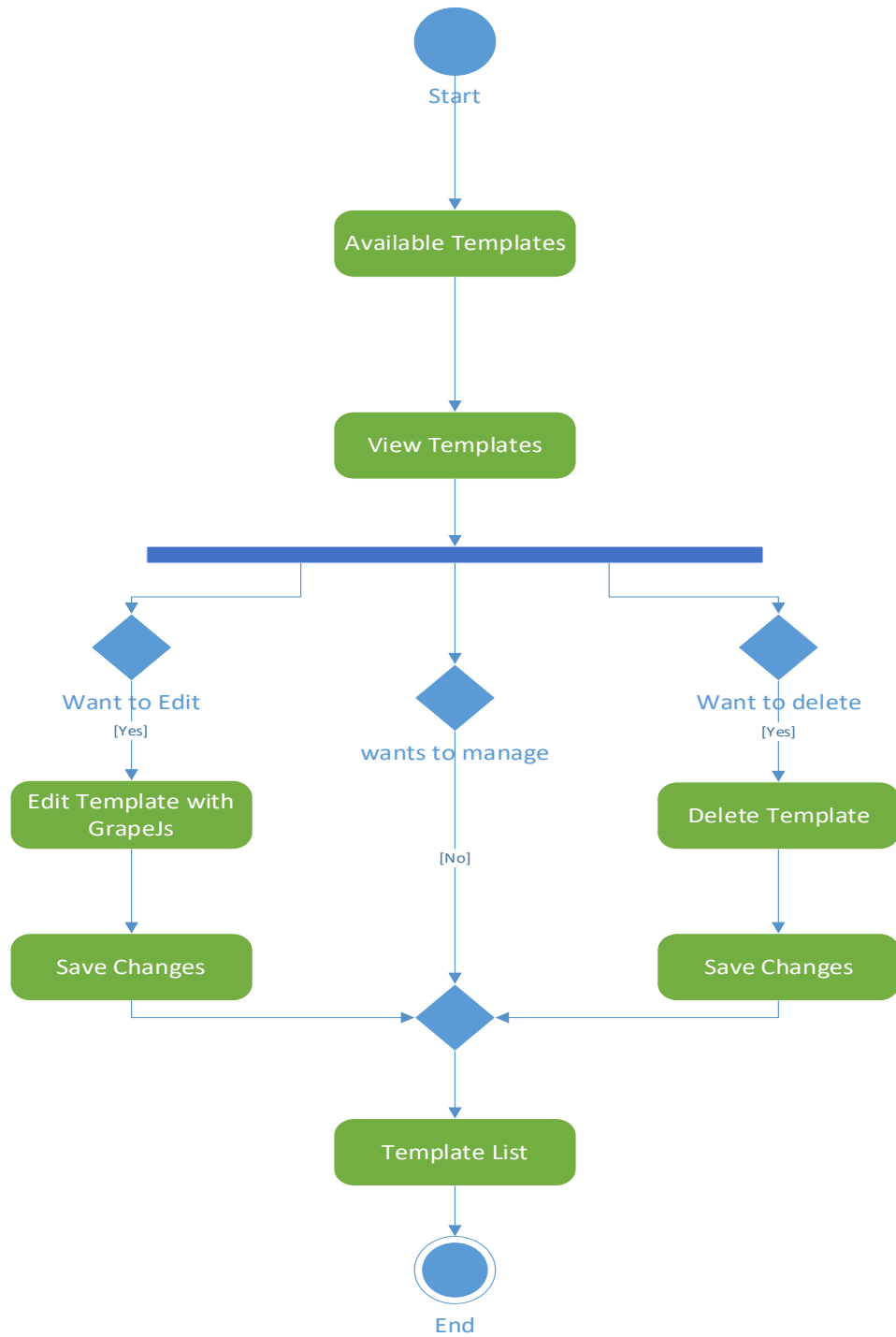


Figure 6: Manage Template

View Prompt history:

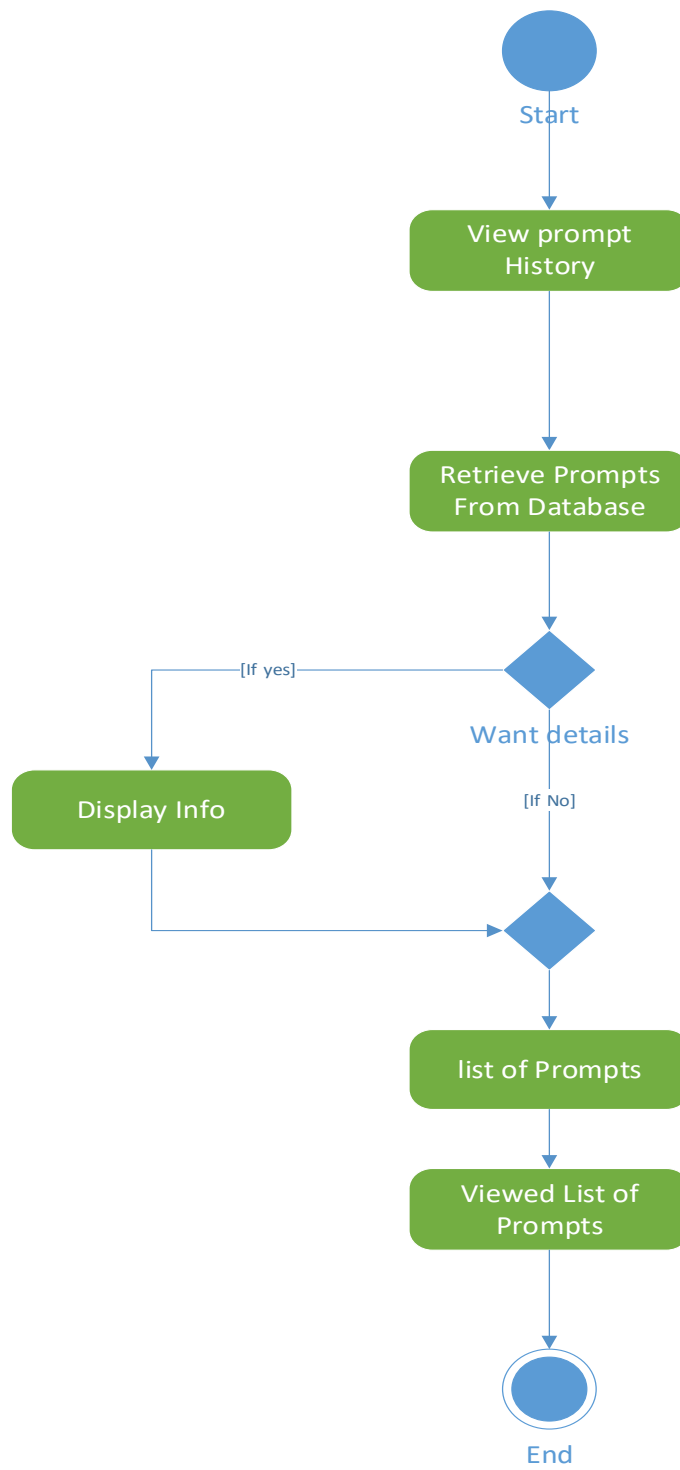


Figure 7: View Prompt History

2.1.2.2 User End

Sign Up:

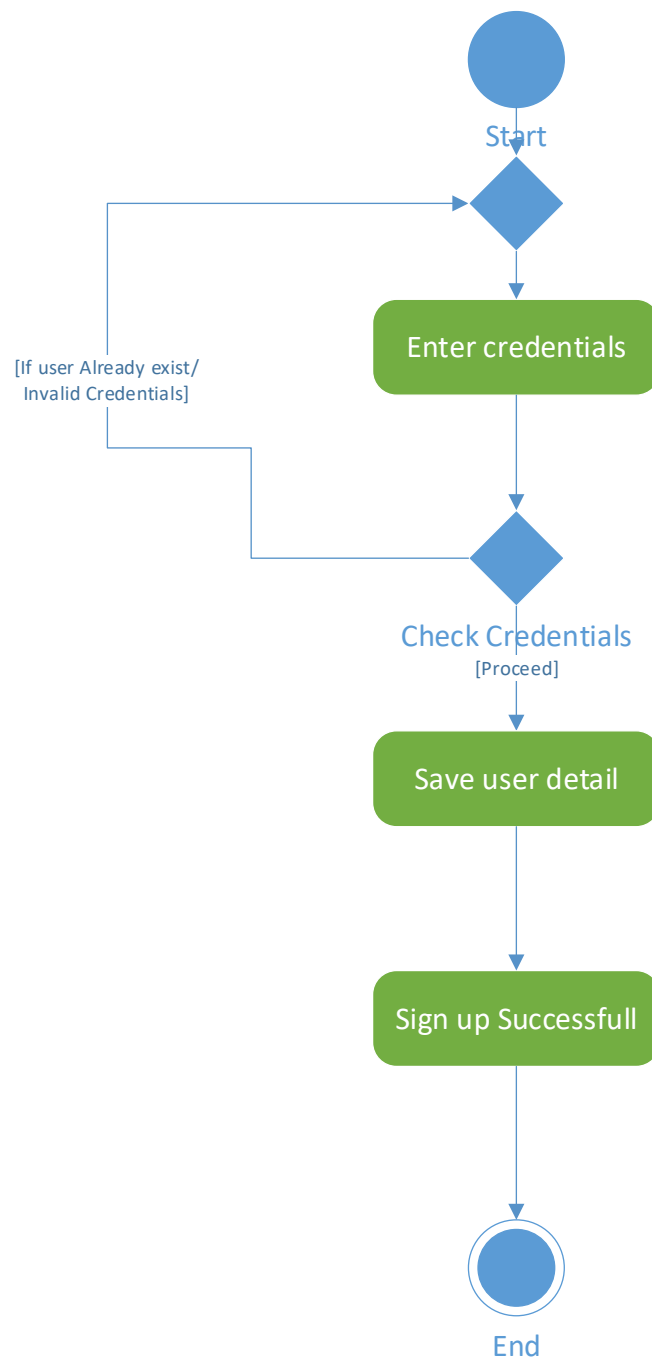


Figure 8: Sign Up

Sign In:

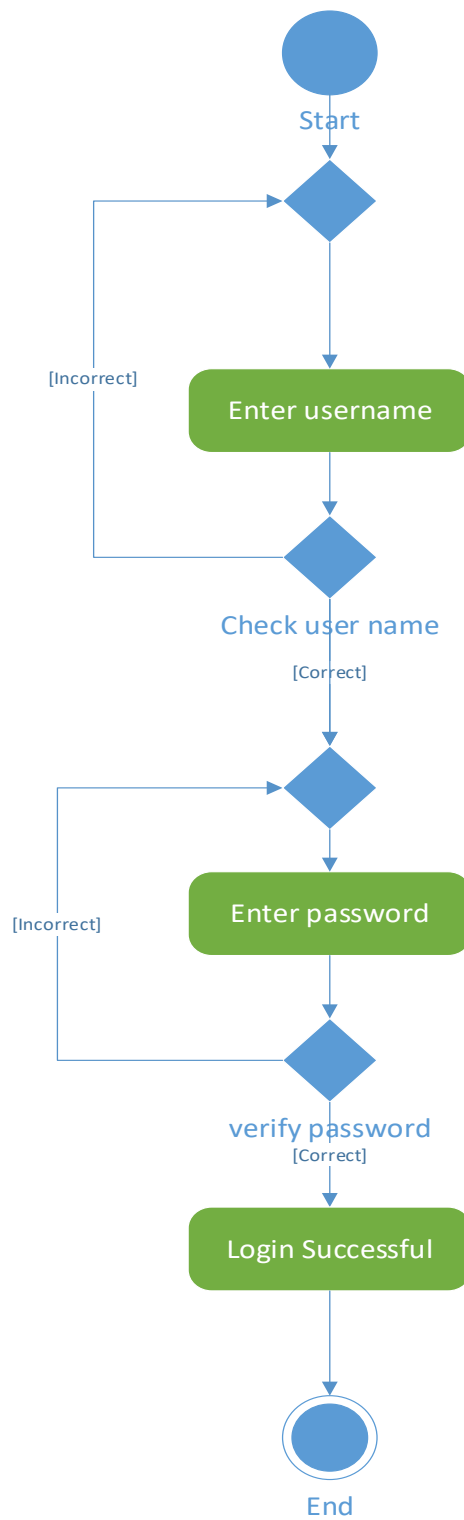


Figure 9: Sign In

Create website:

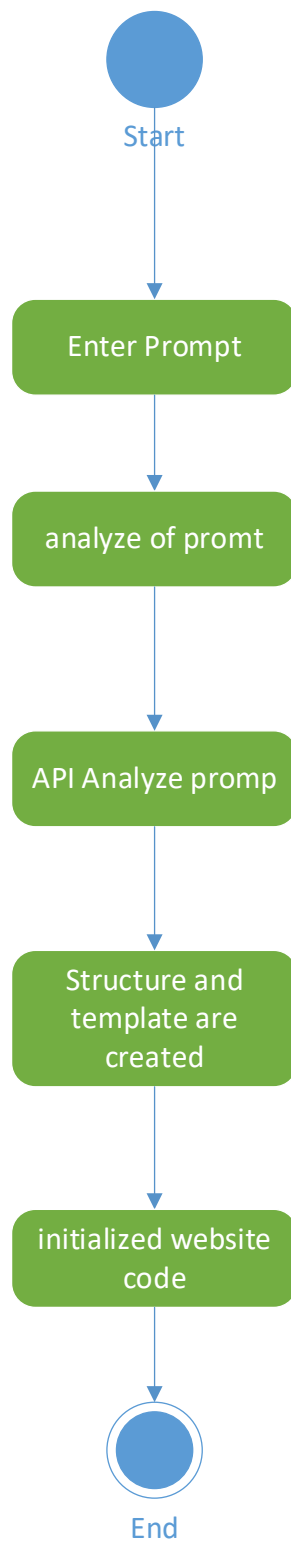


Figure 10: Create Website

GrapeJs Editing:

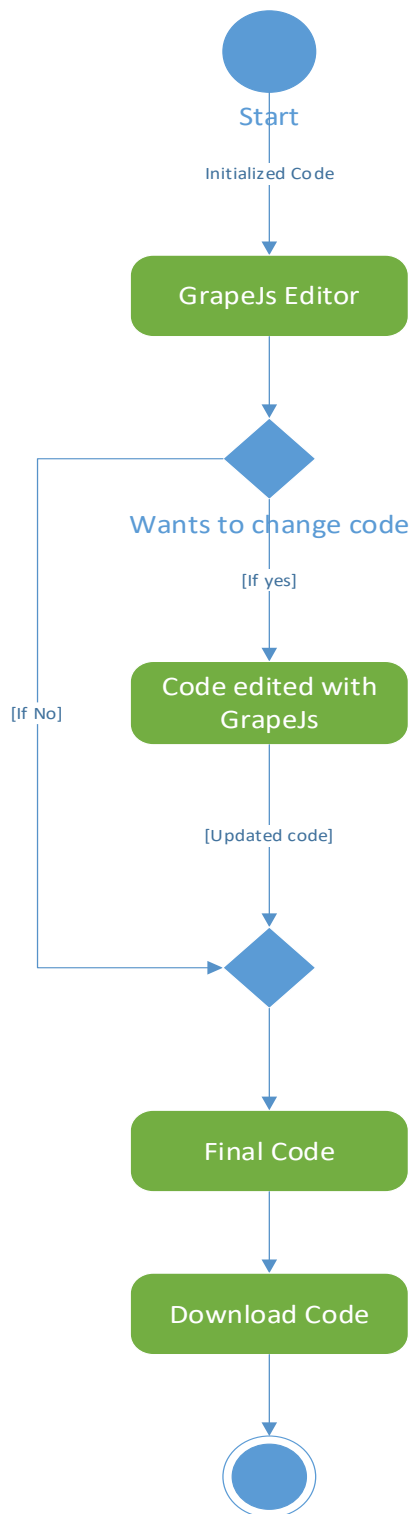


Figure 11: GrapeJs Editing

View Features:

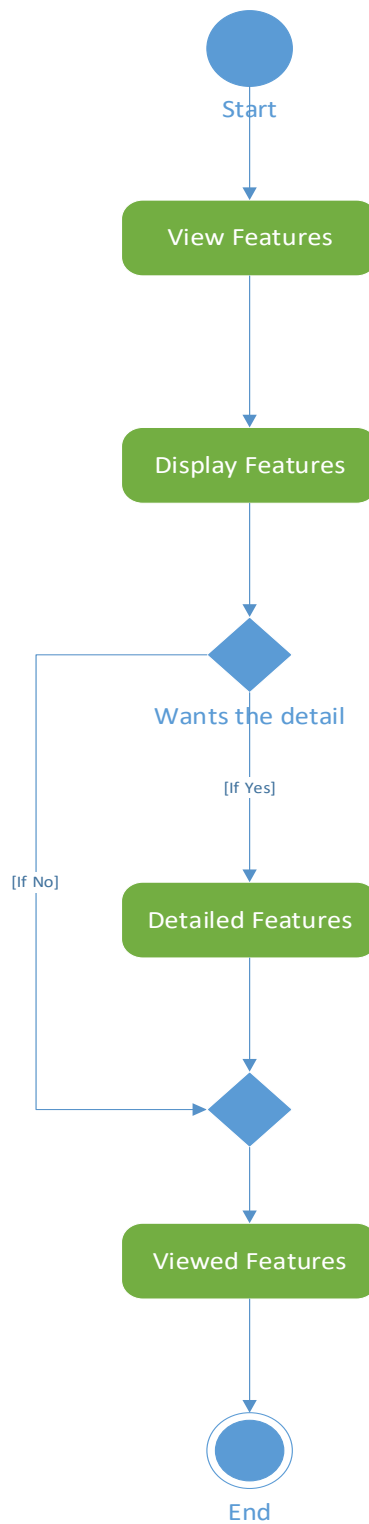


Figure 12: View Features

Contact Us:

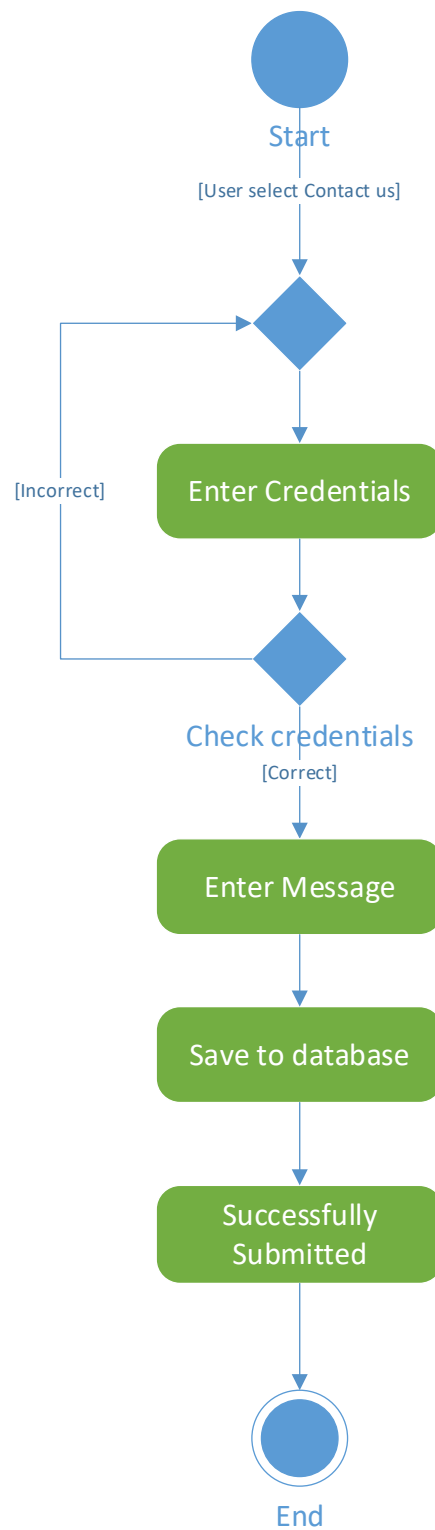


Figure 13: Contact Us

2.1.3 State Diagram

2.1.3.1 Admin End

Sign Up:

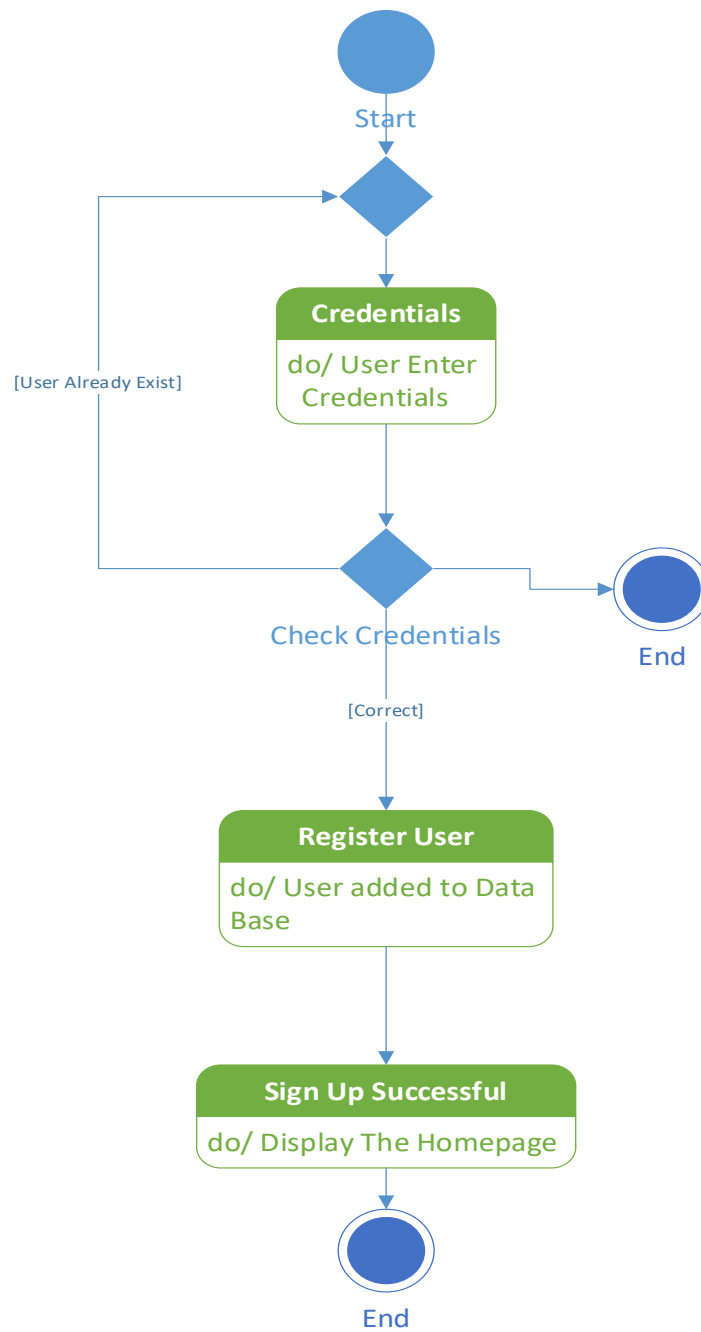


Figure 14: Sign Up

Sign In:

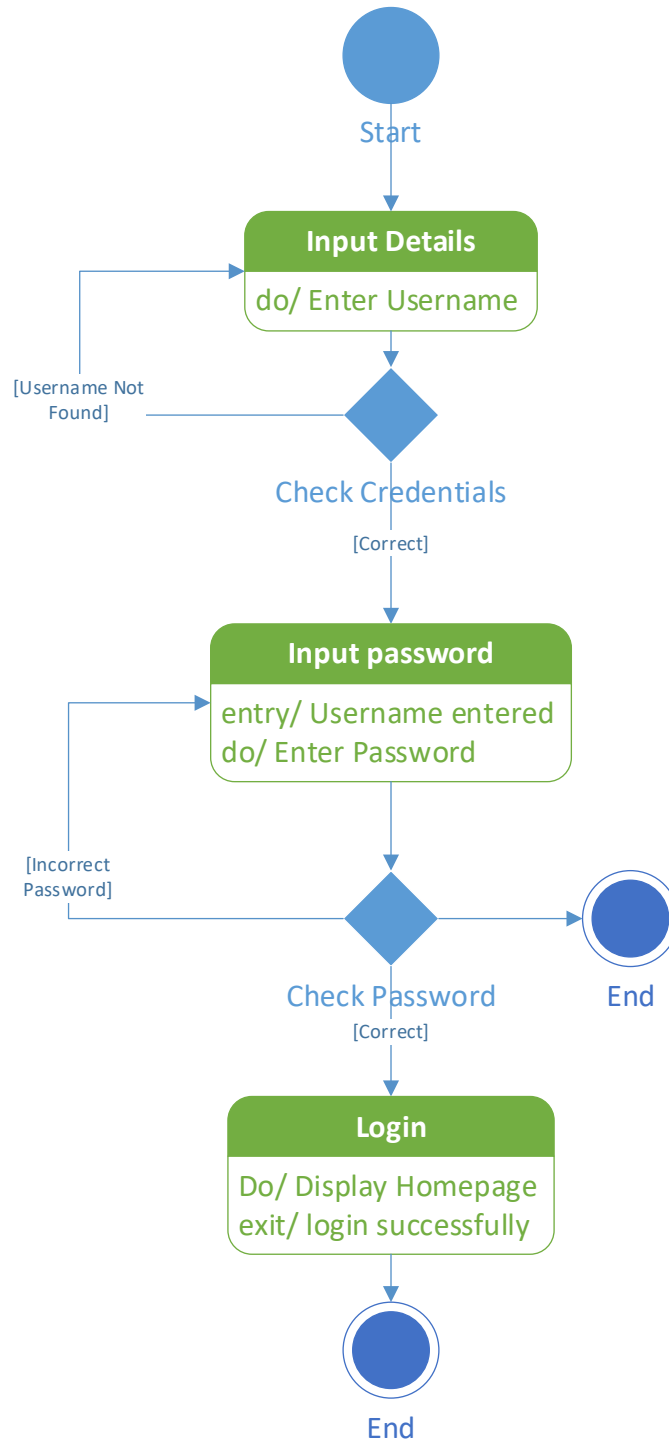


Figure 15: Sign In

Manage Profile:

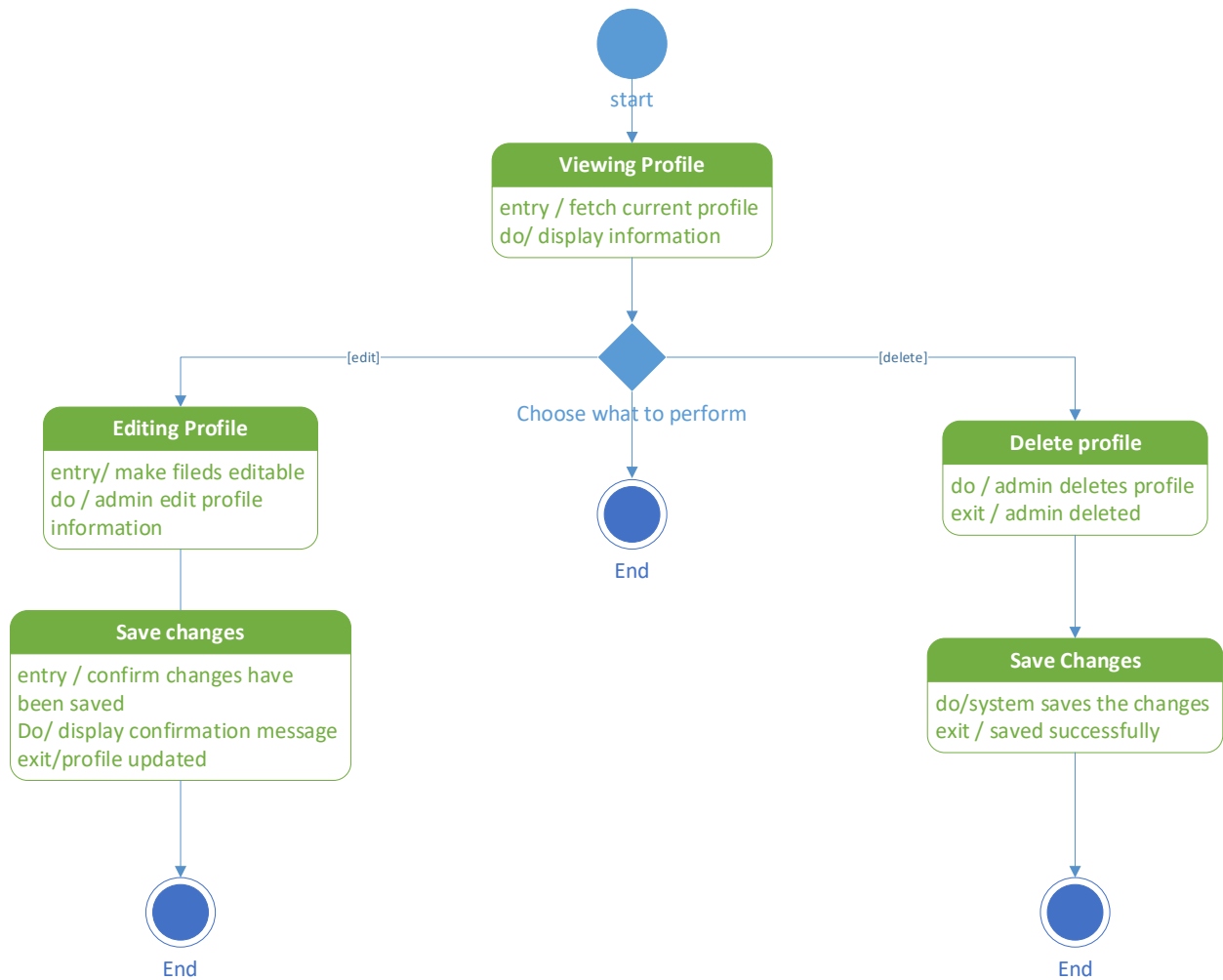


Figure 16: Manage Profile

Manage Template:

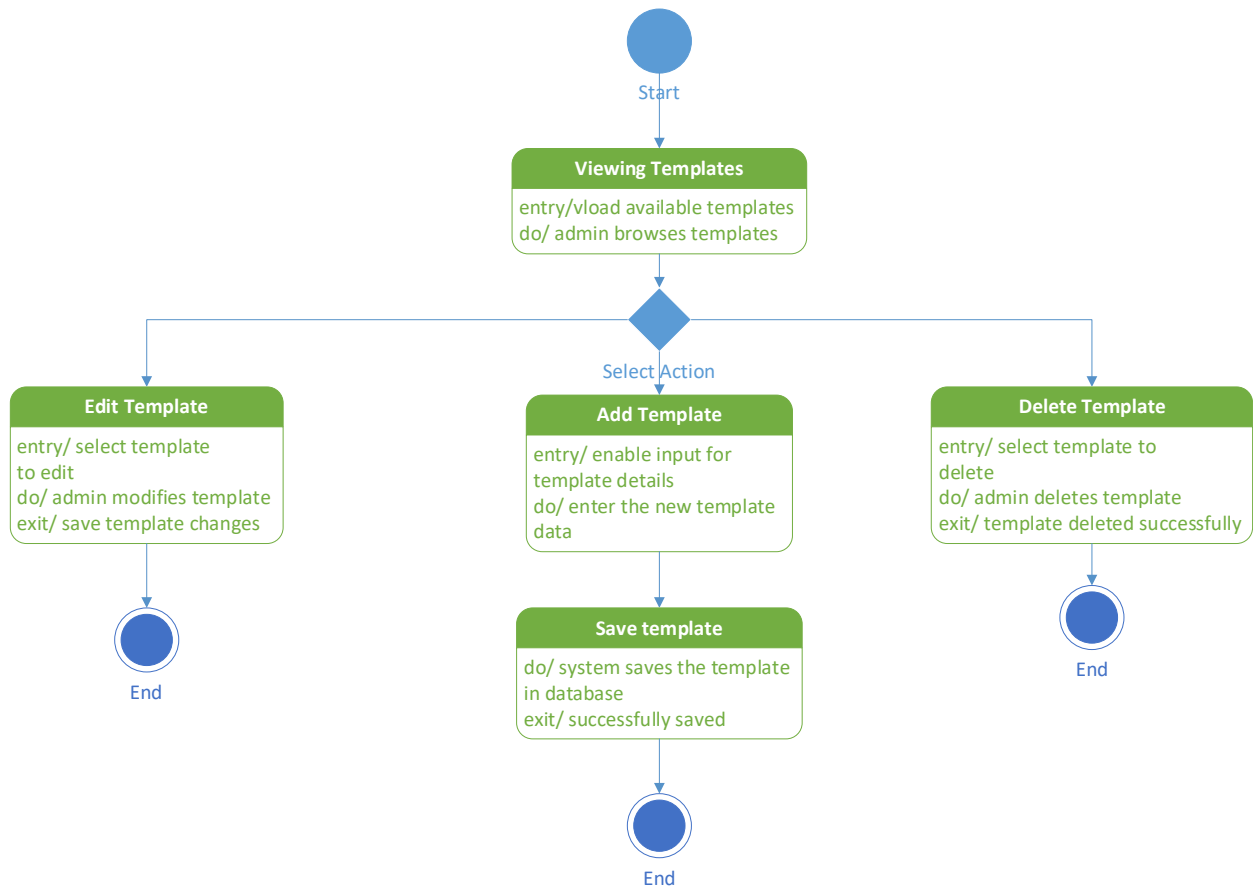


Figure 17: manage Template

View prompt history:

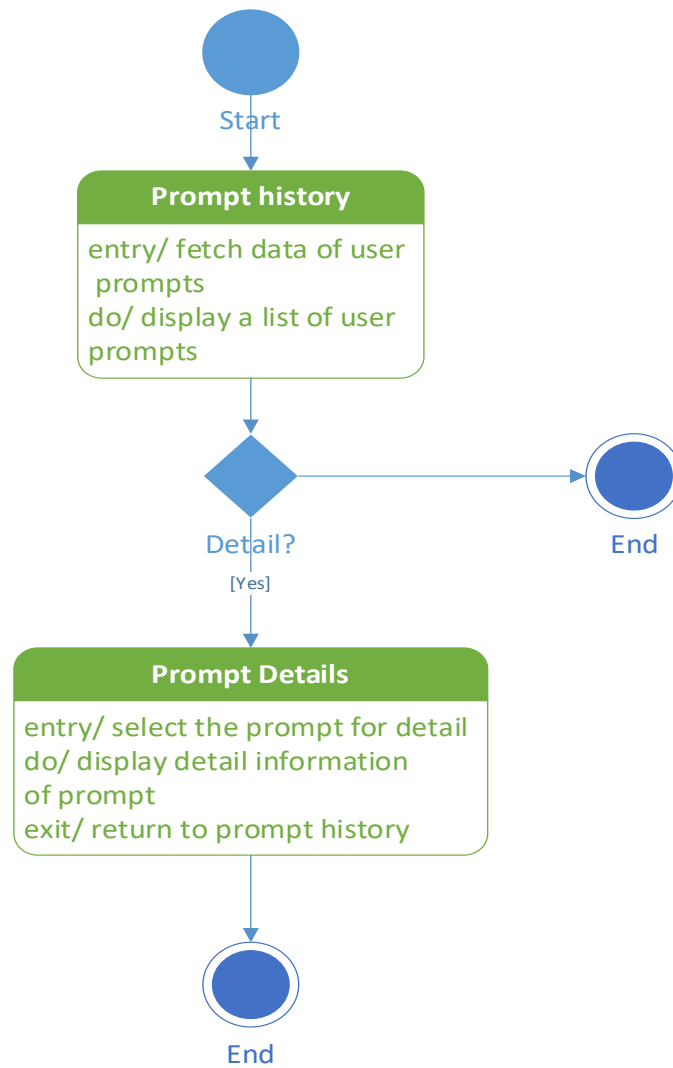


Figure 18: View Prompt History

2.1.3.2 User End

Sign Up:

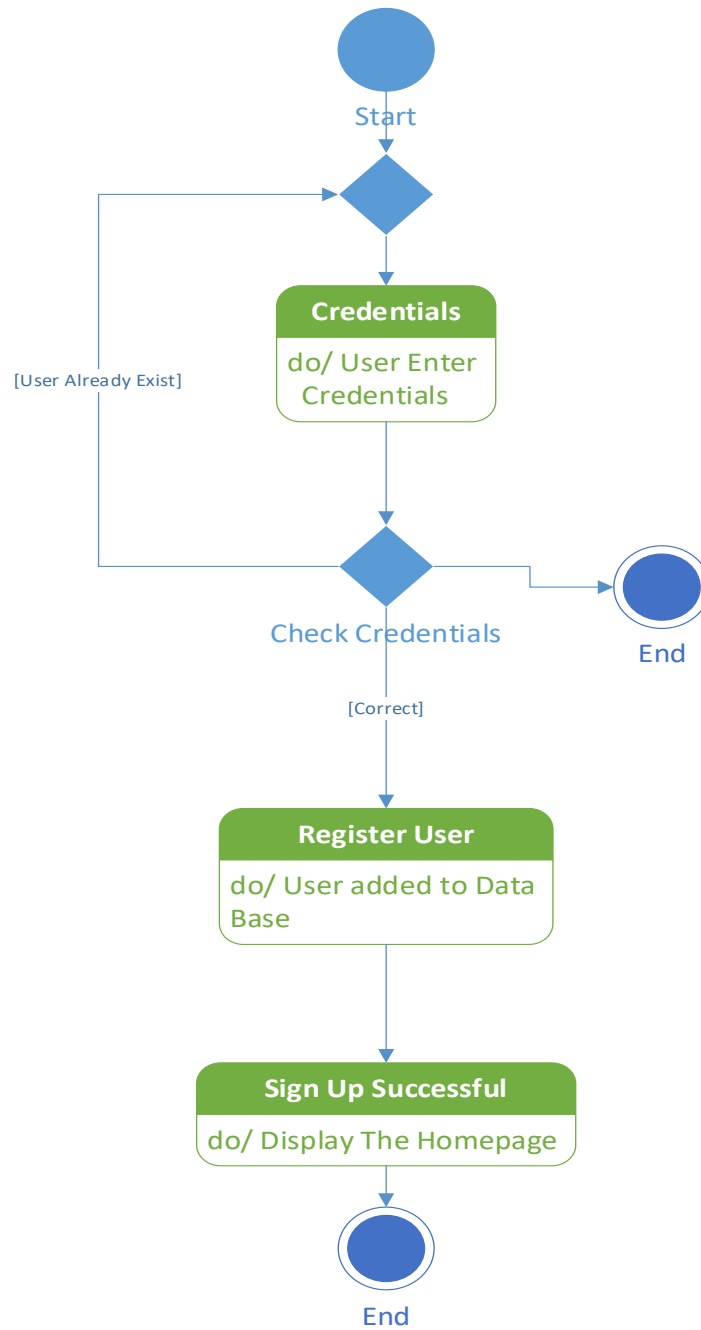


Figure 19: Sign Up

Sign In:

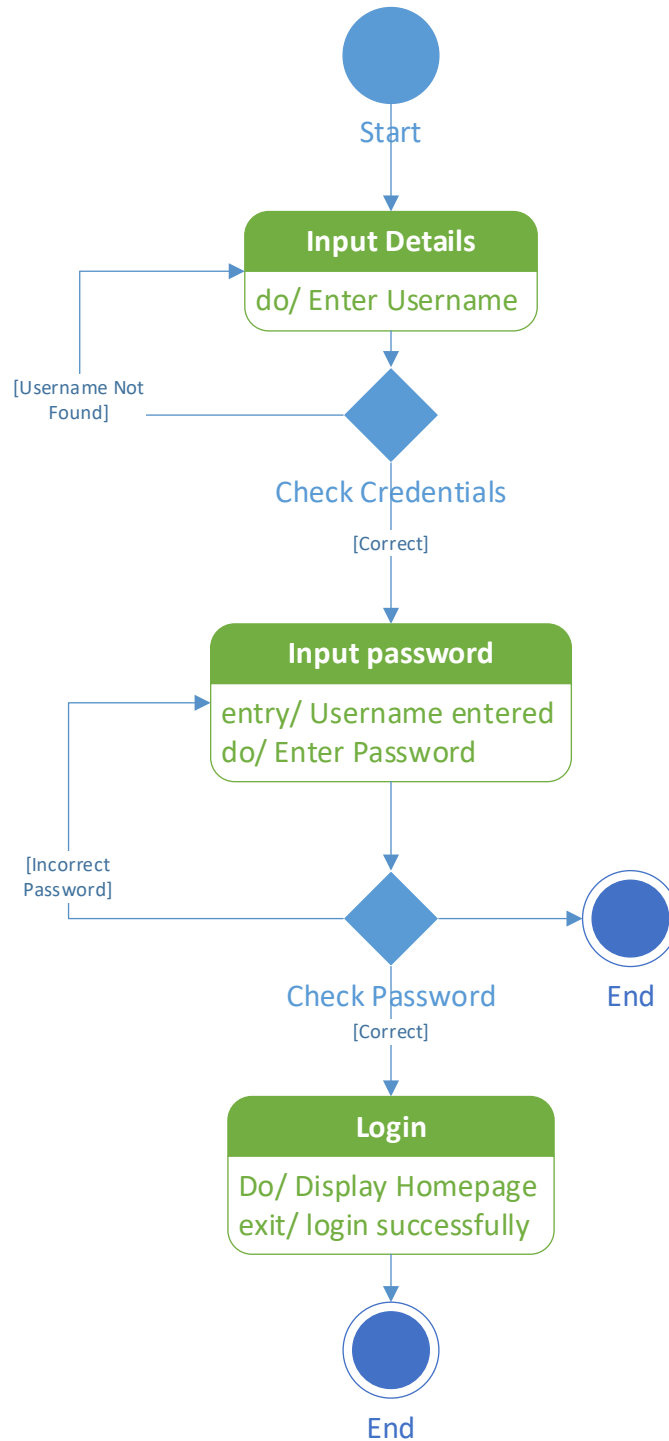


Figure 20: Sign In

Create Website:

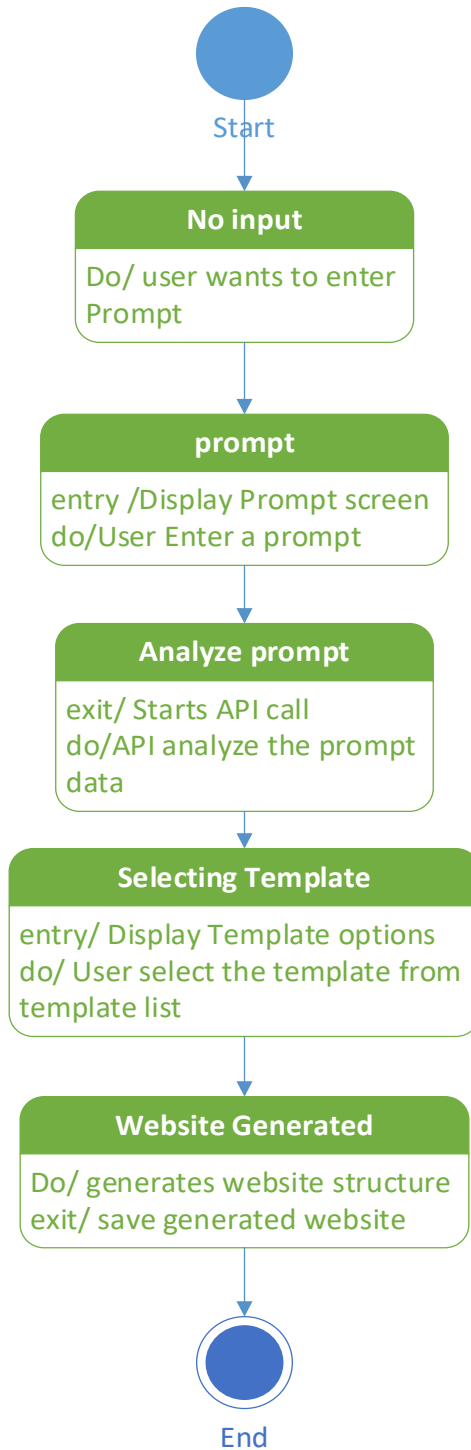


Figure 21: Create Website

GrapeJs Editing:

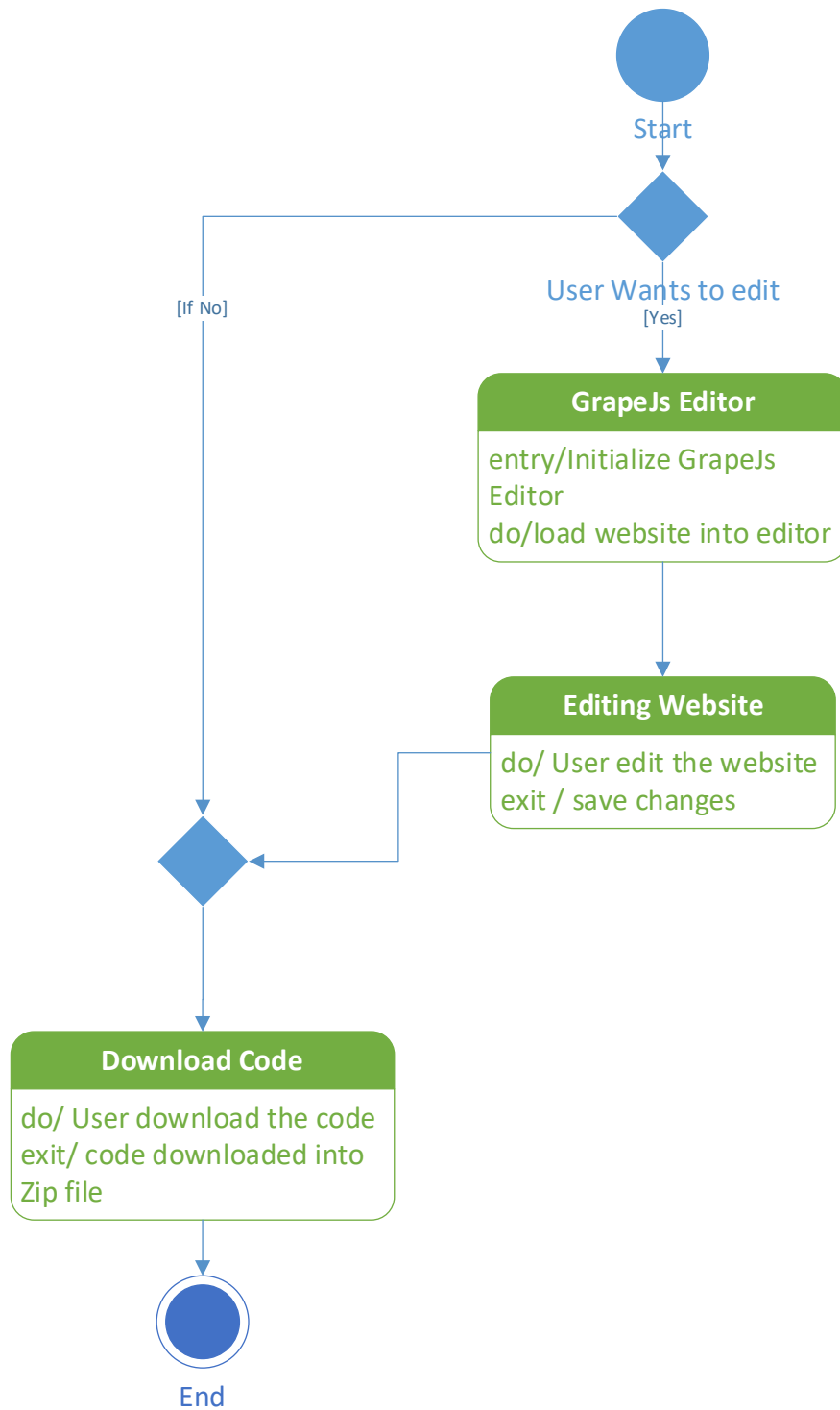


Figure 22: GrapeJs Editing

View Features:

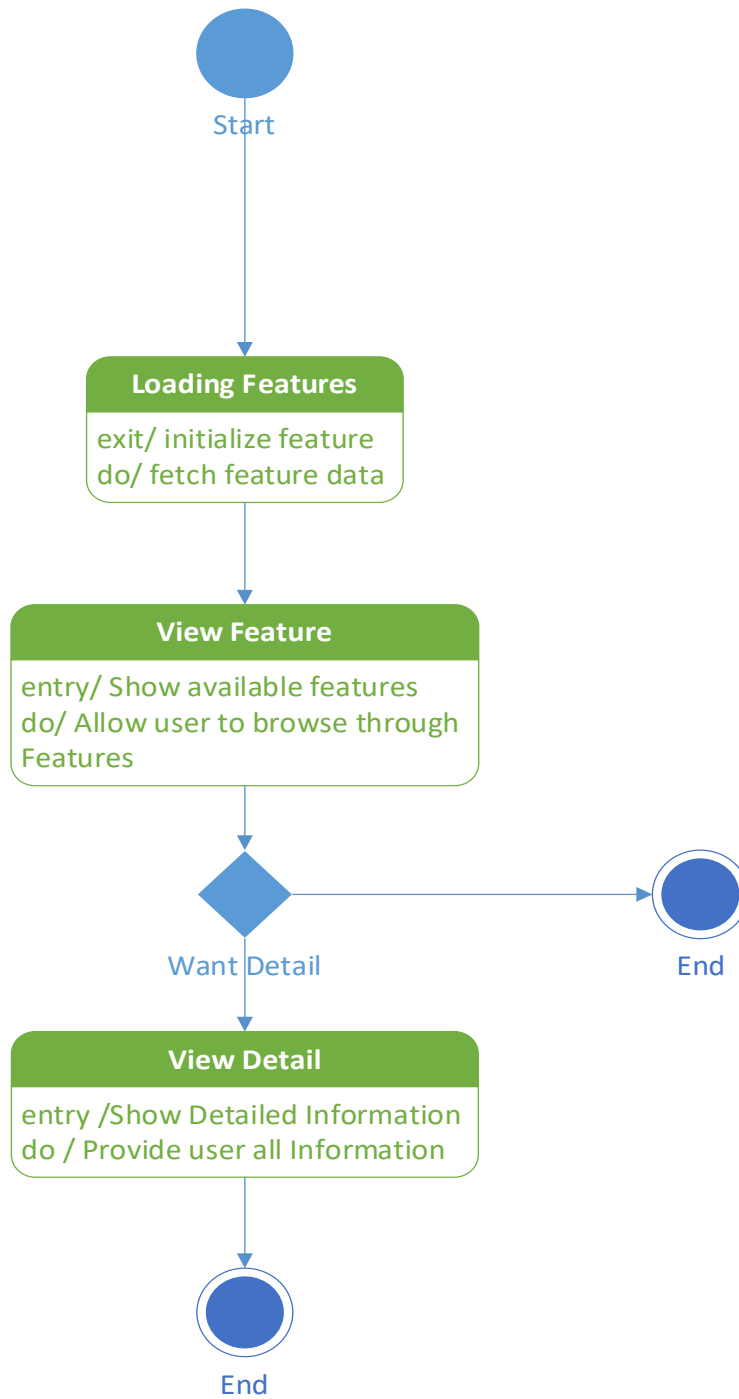


Figure 23: View Features

Contact Us:

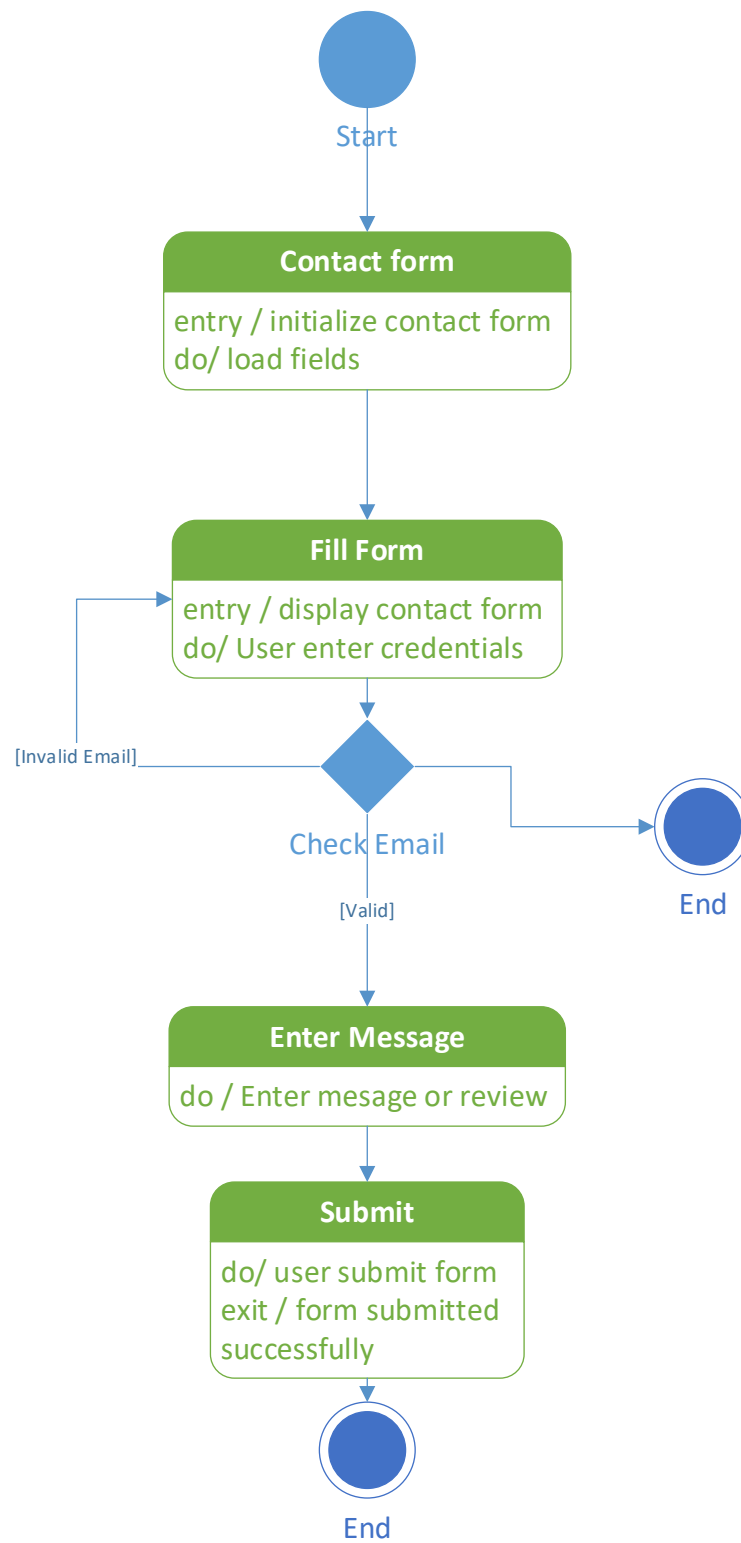


Figure 24: Contact Us

2.1.4 Sequence Diagram

2.1.4.1 Admin End

Sign Up:

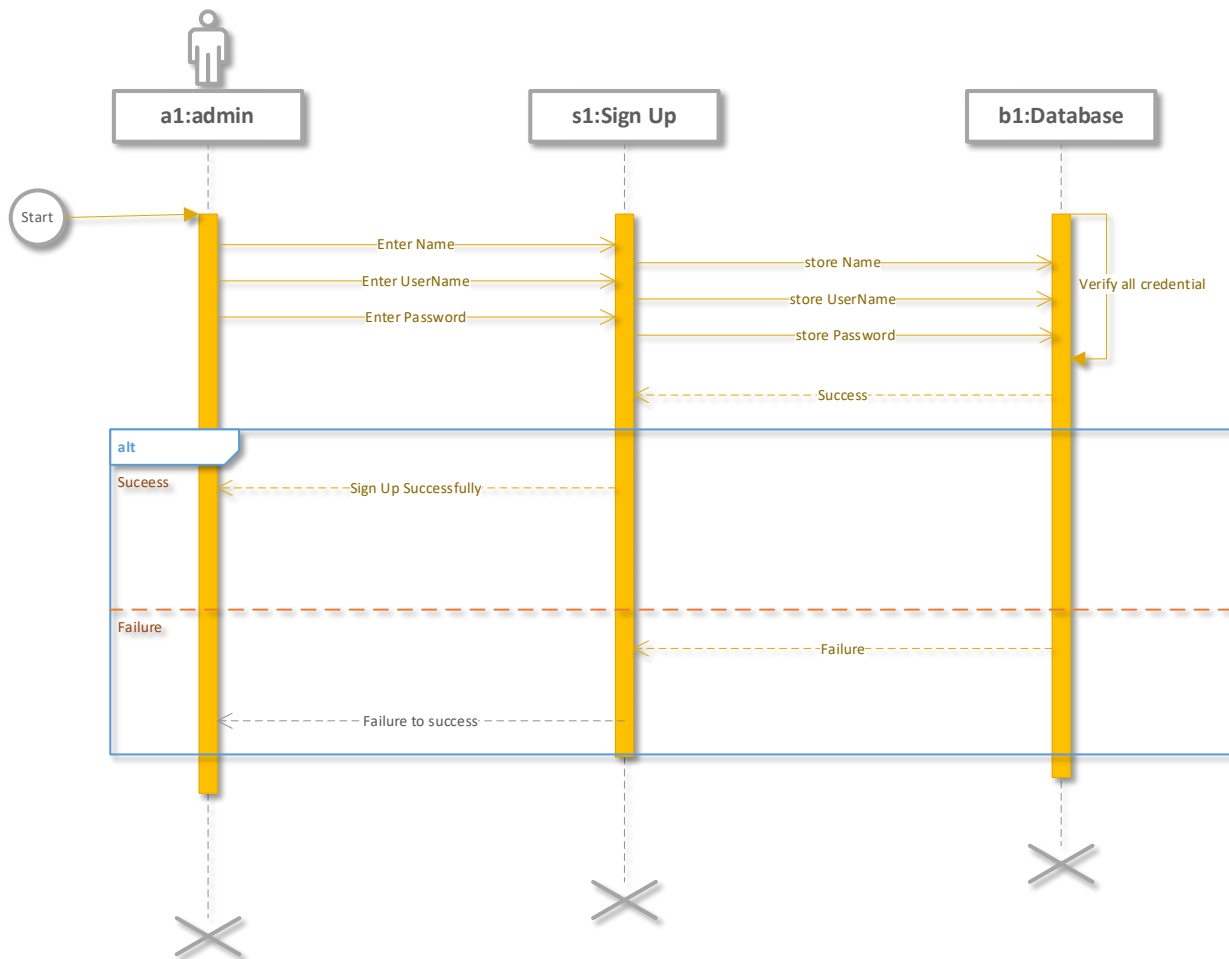
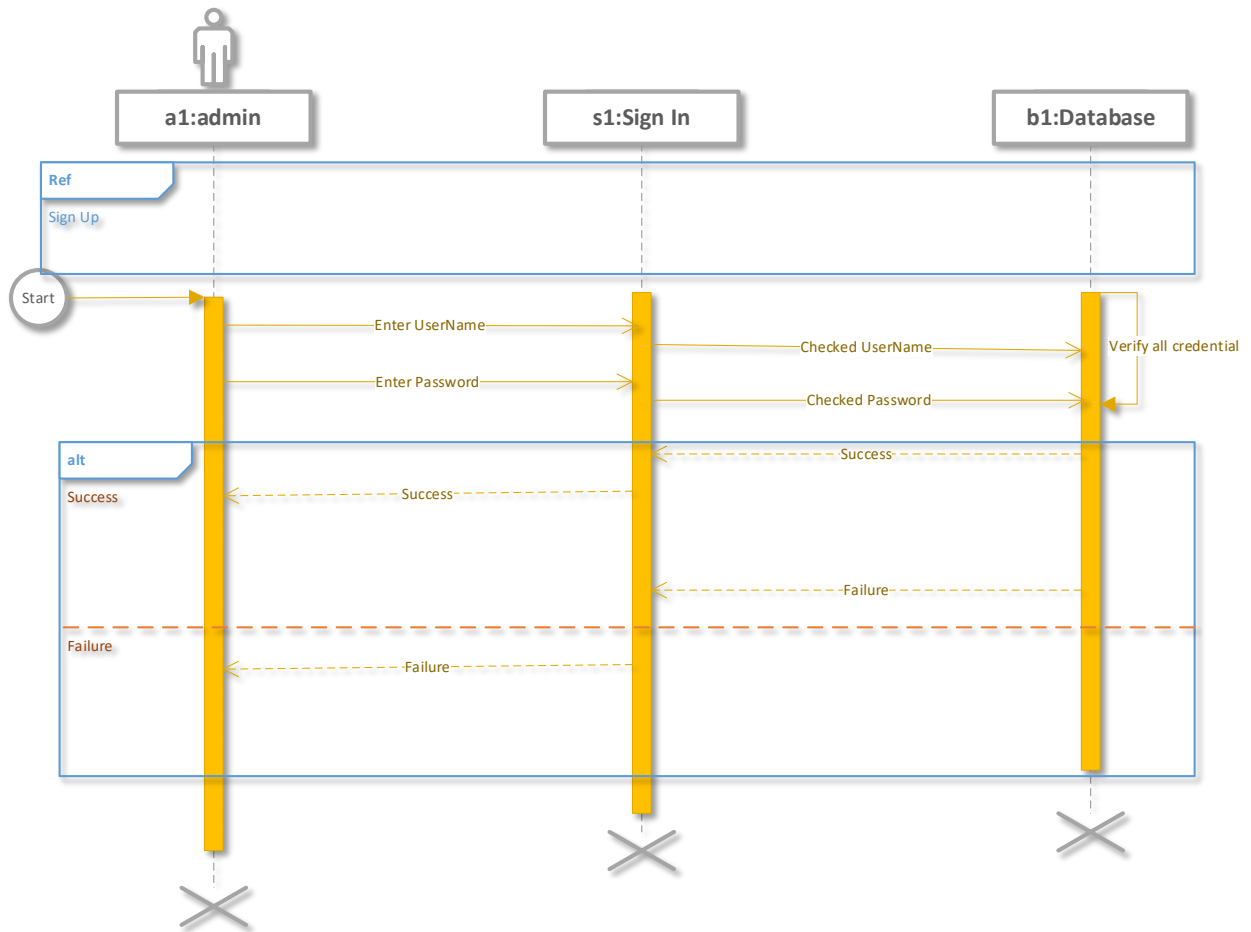
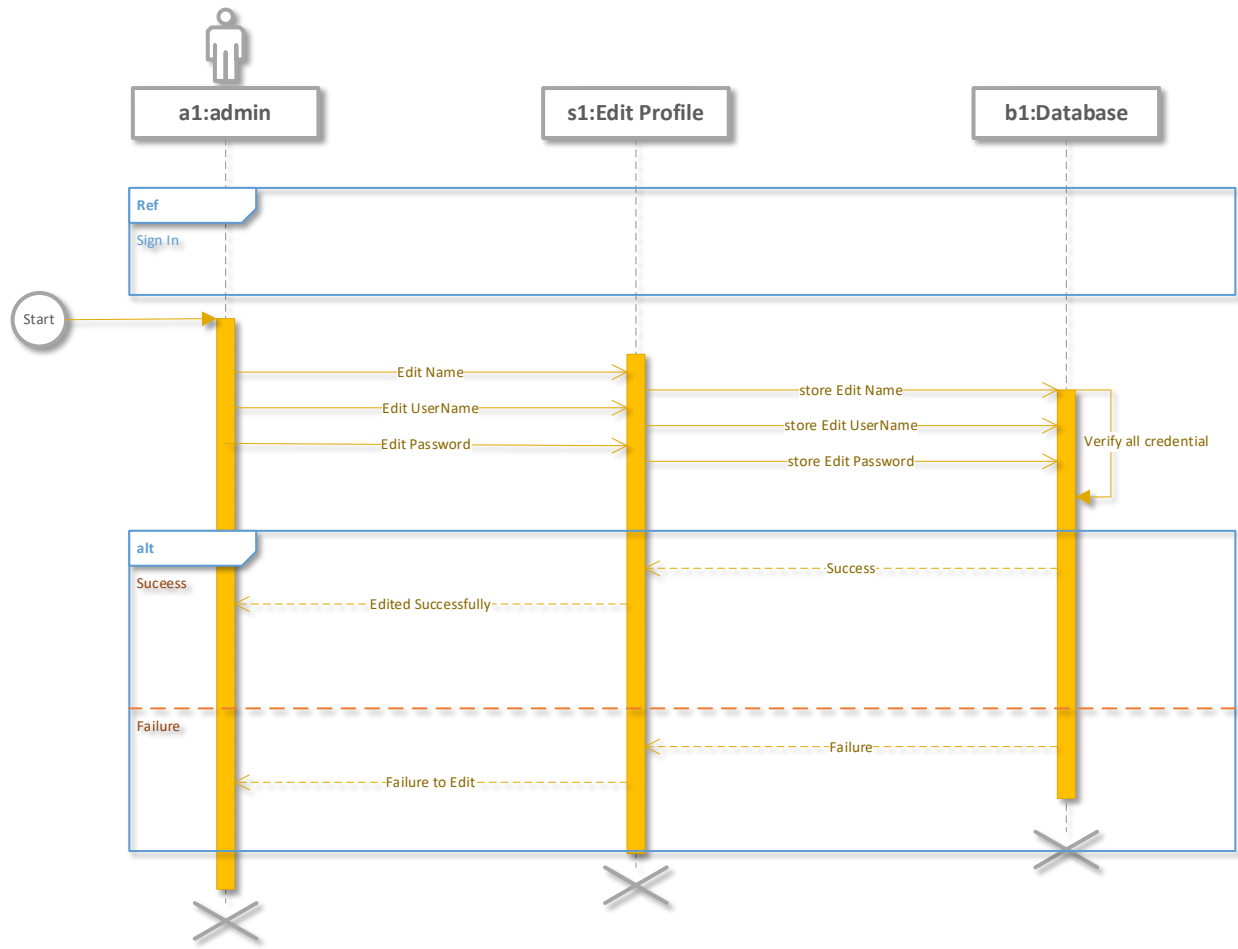


Figure 25: Admin End

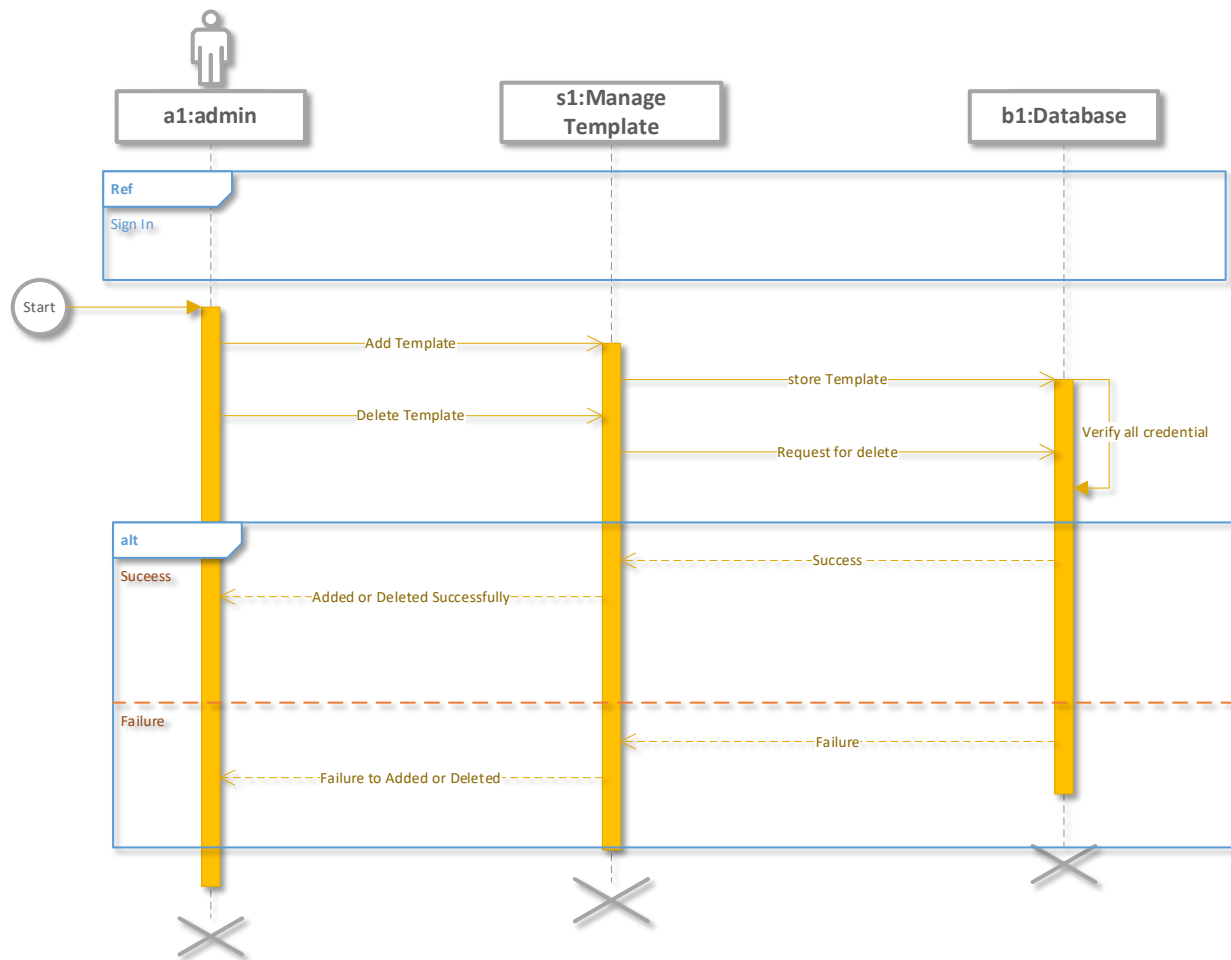
Sign In



Edit Profile:



Manage Template:



2.1.4.2 User End

Sign Up:

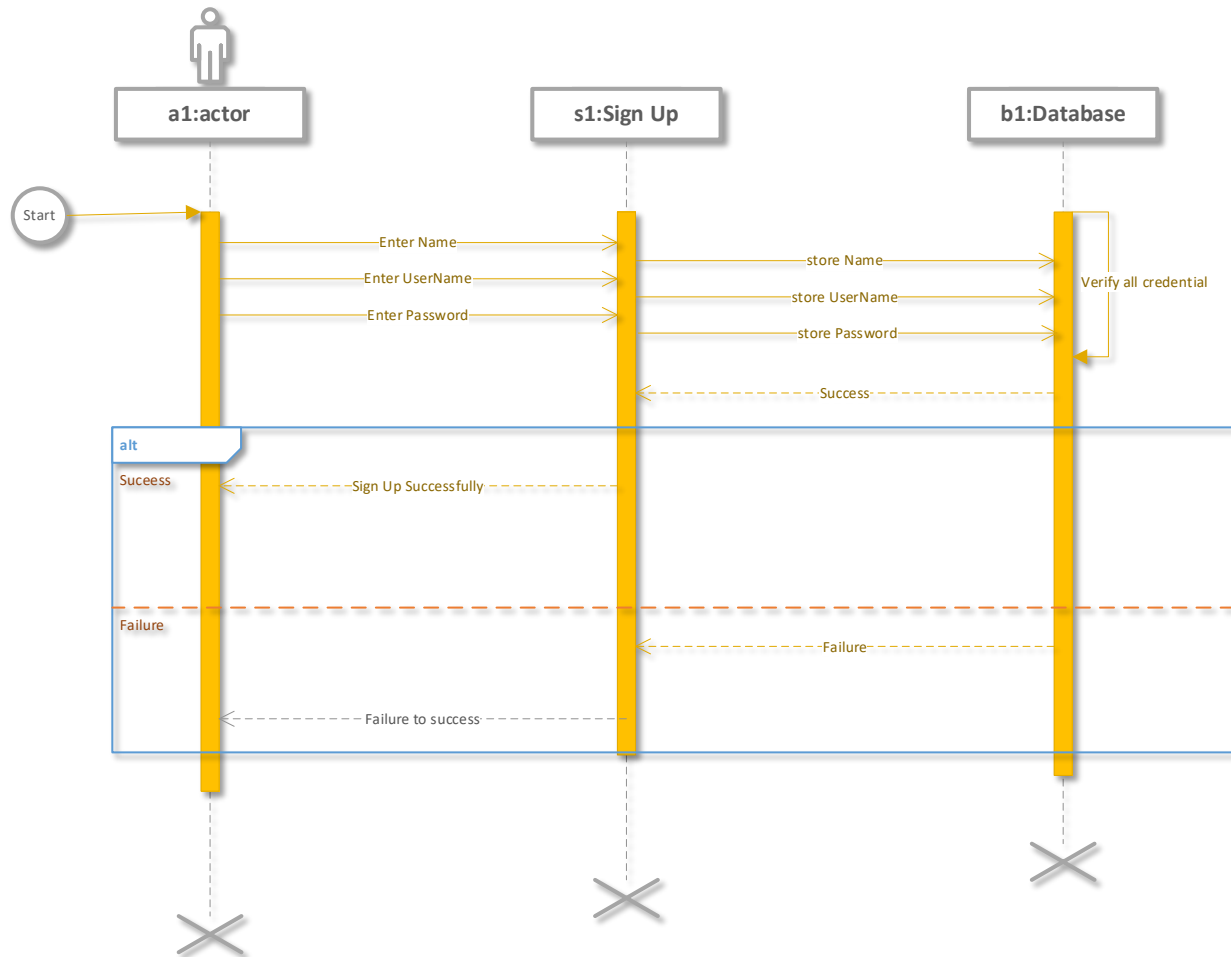
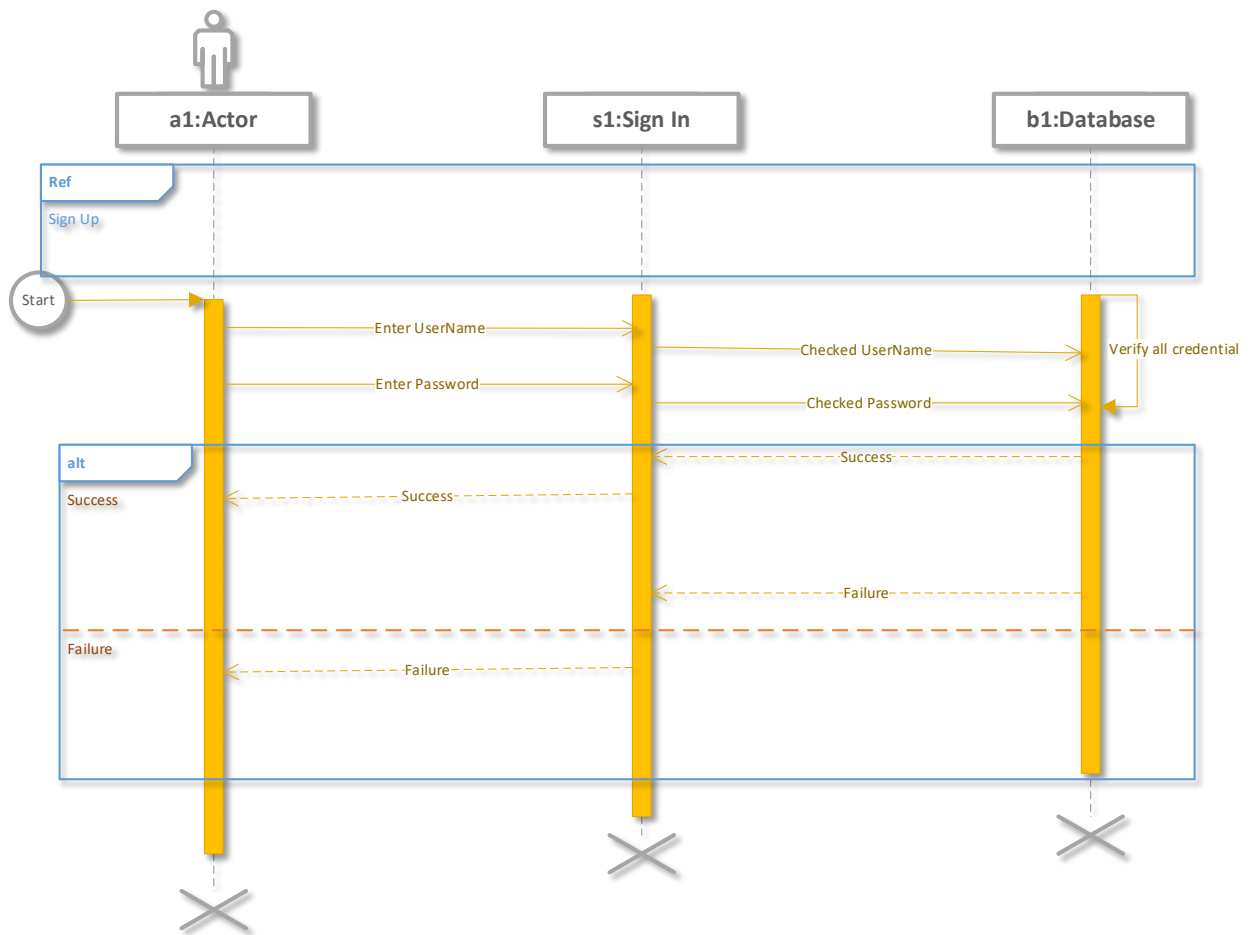
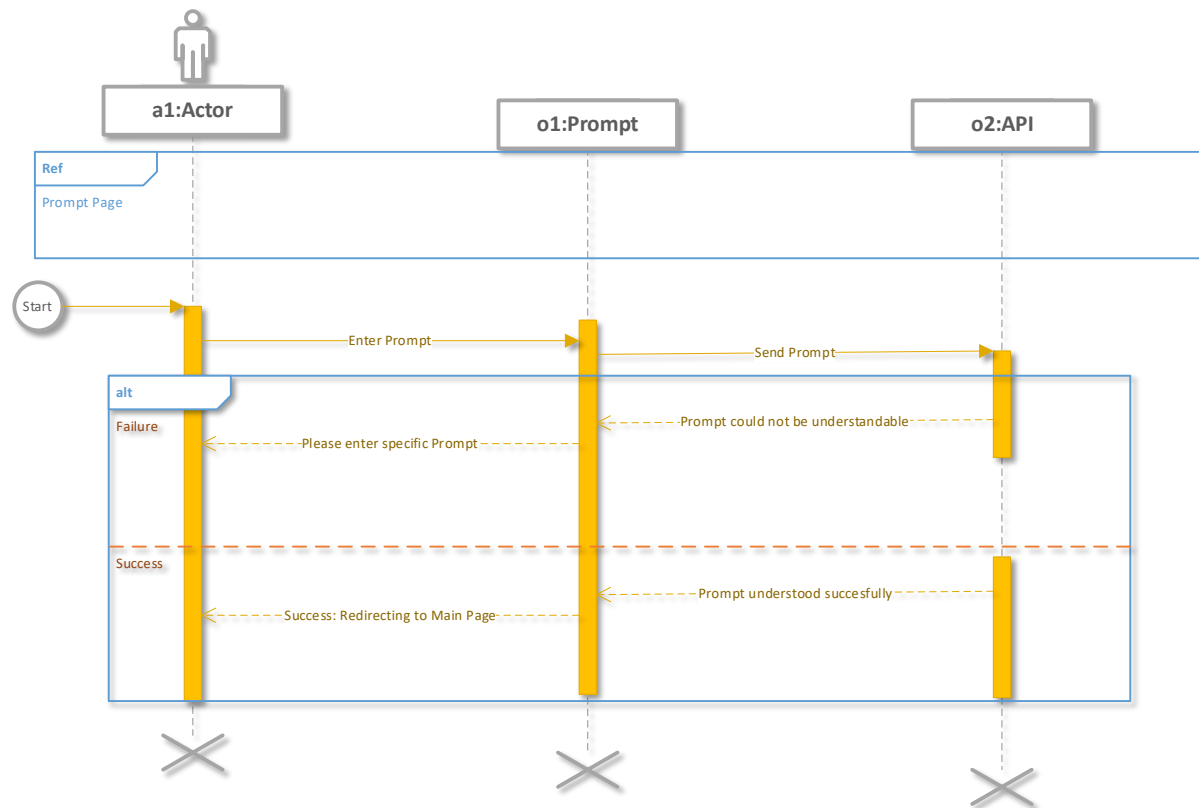
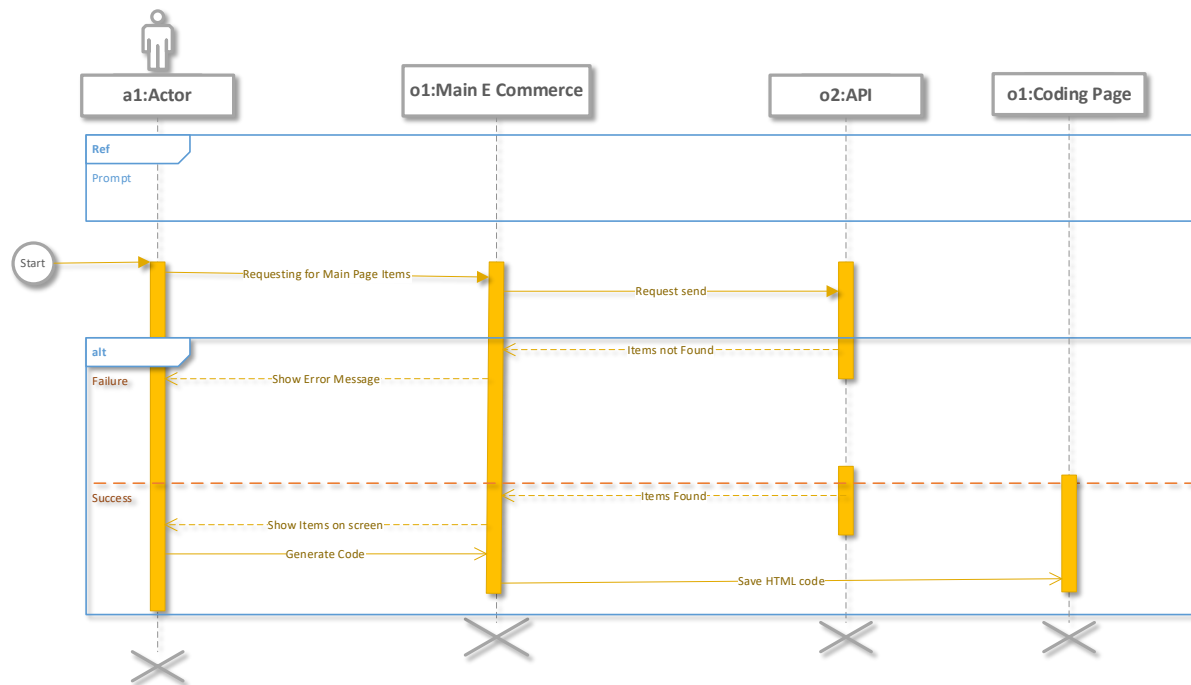


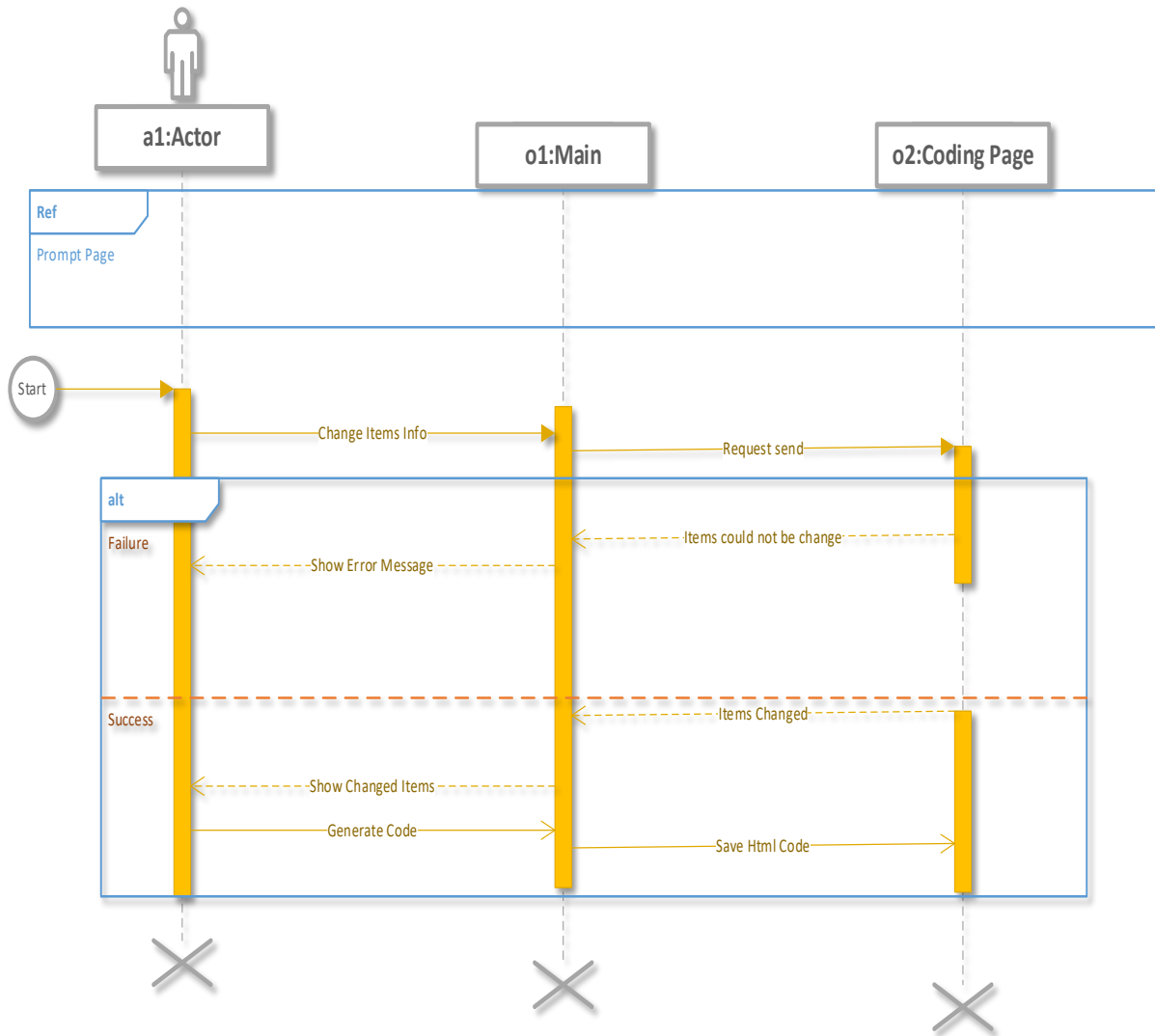
Figure 26: User End

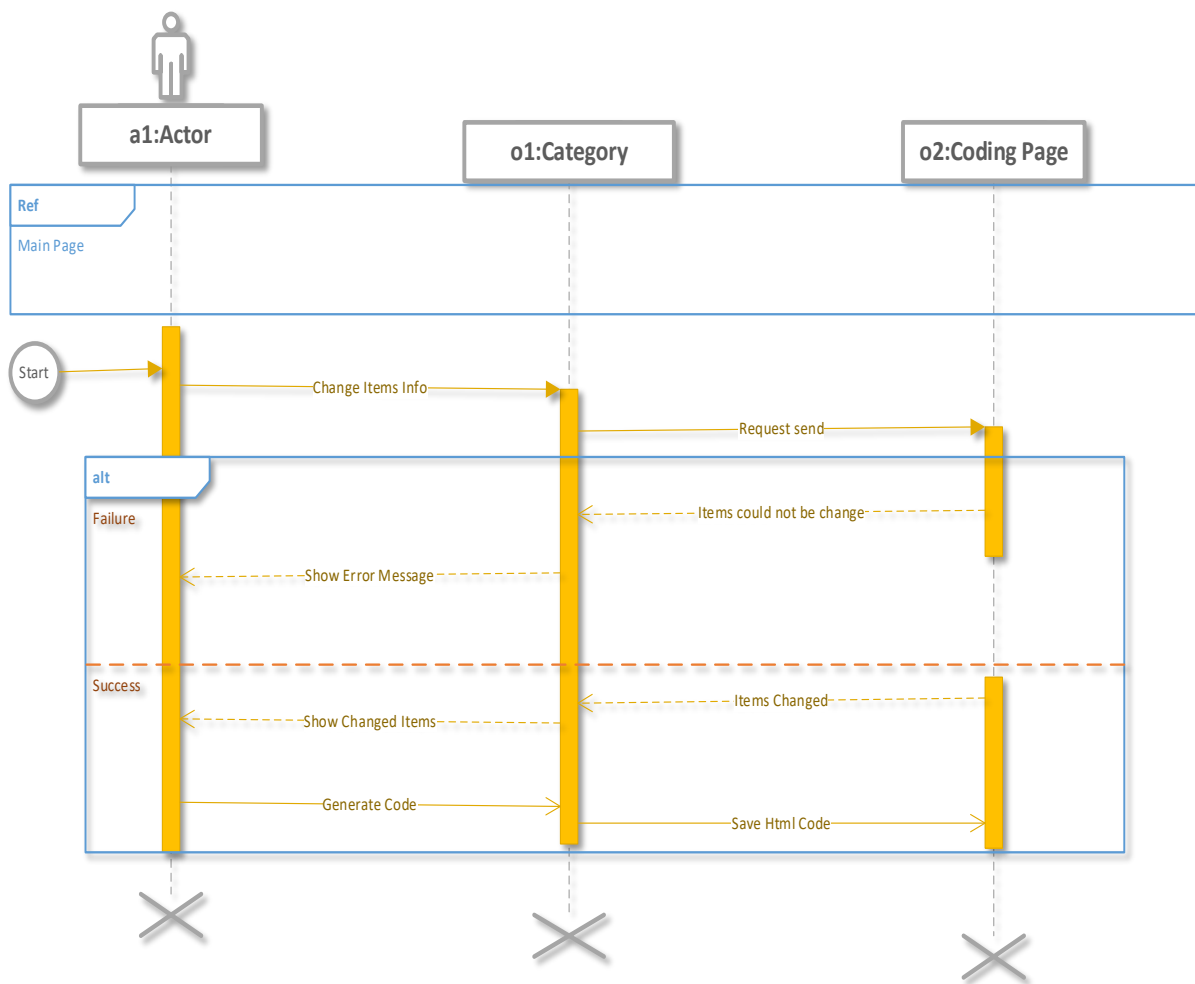
Sign In

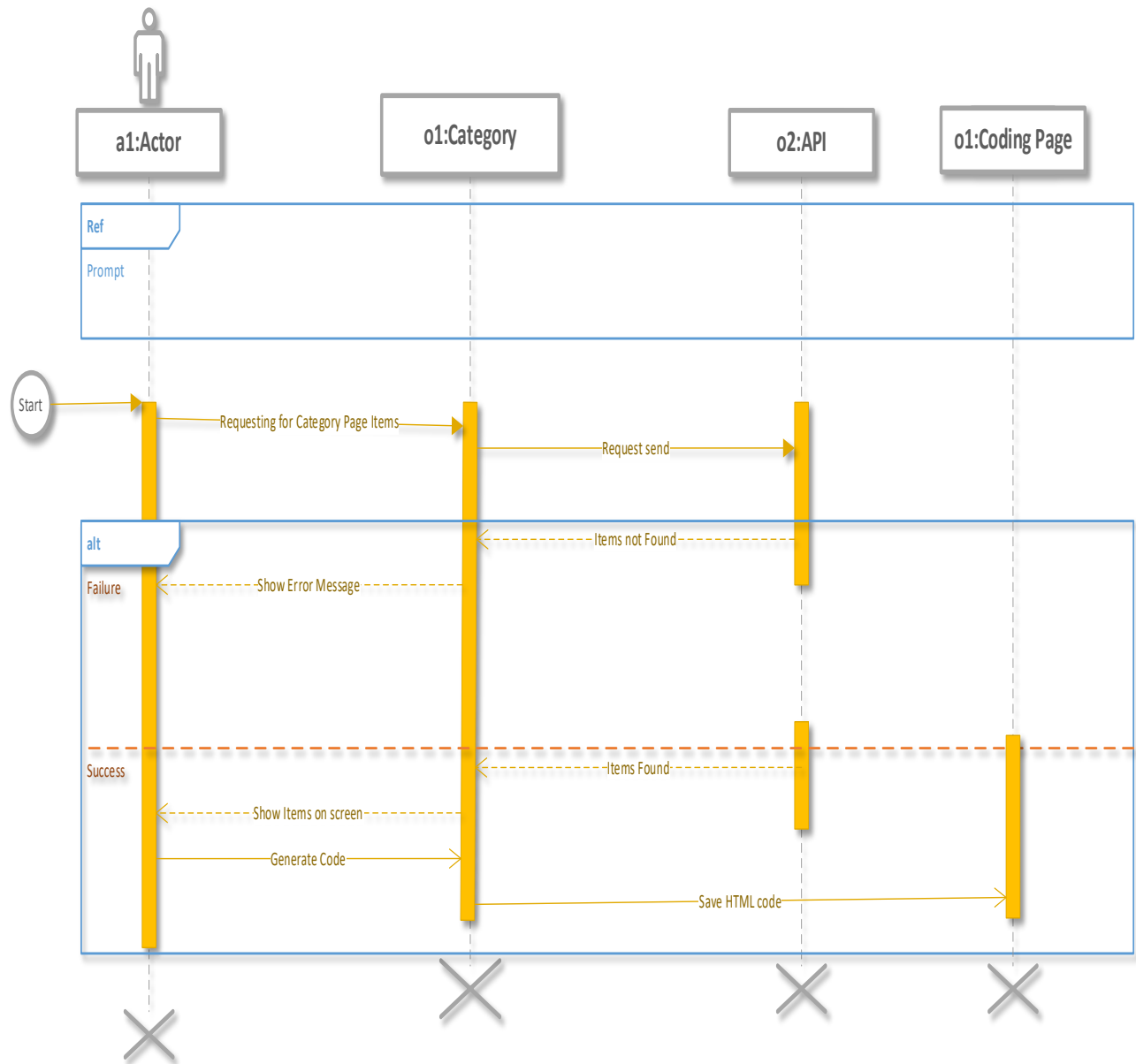


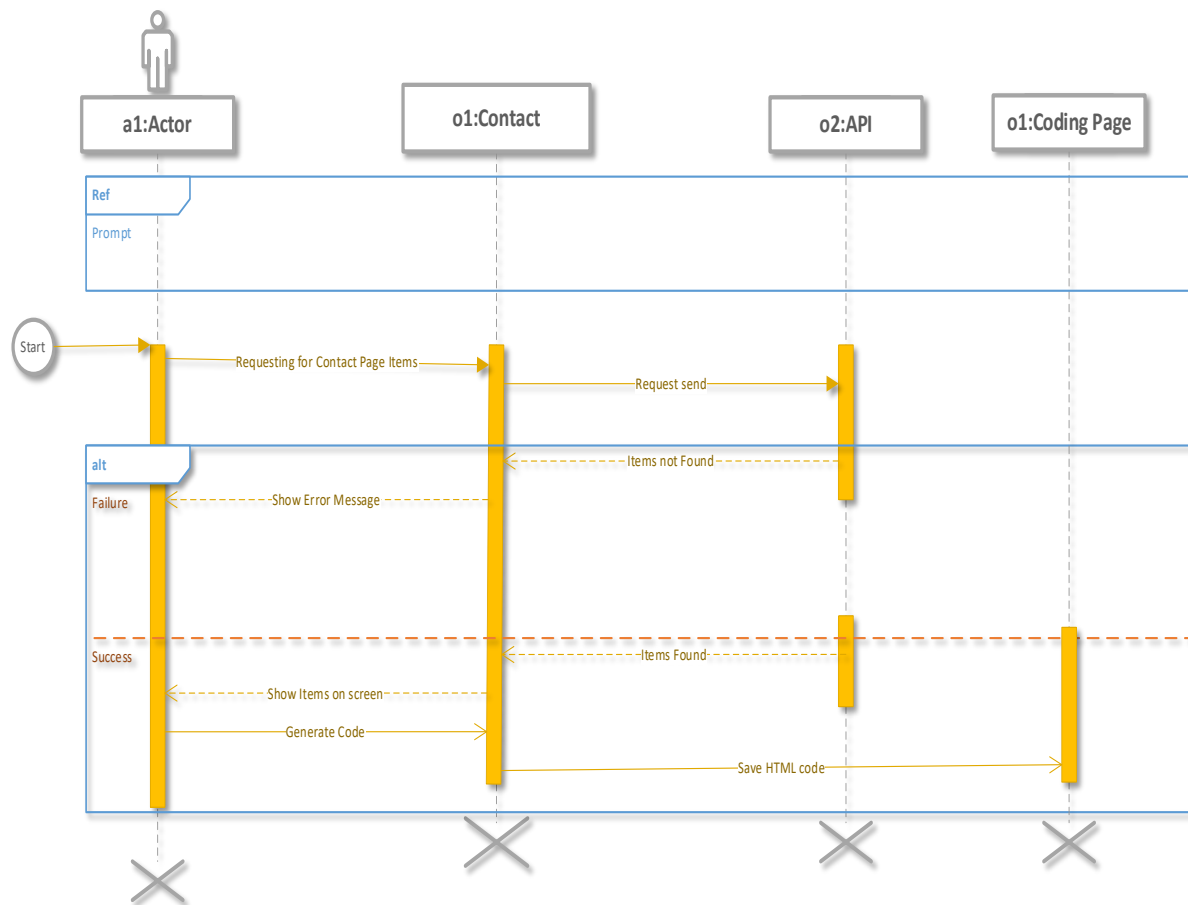


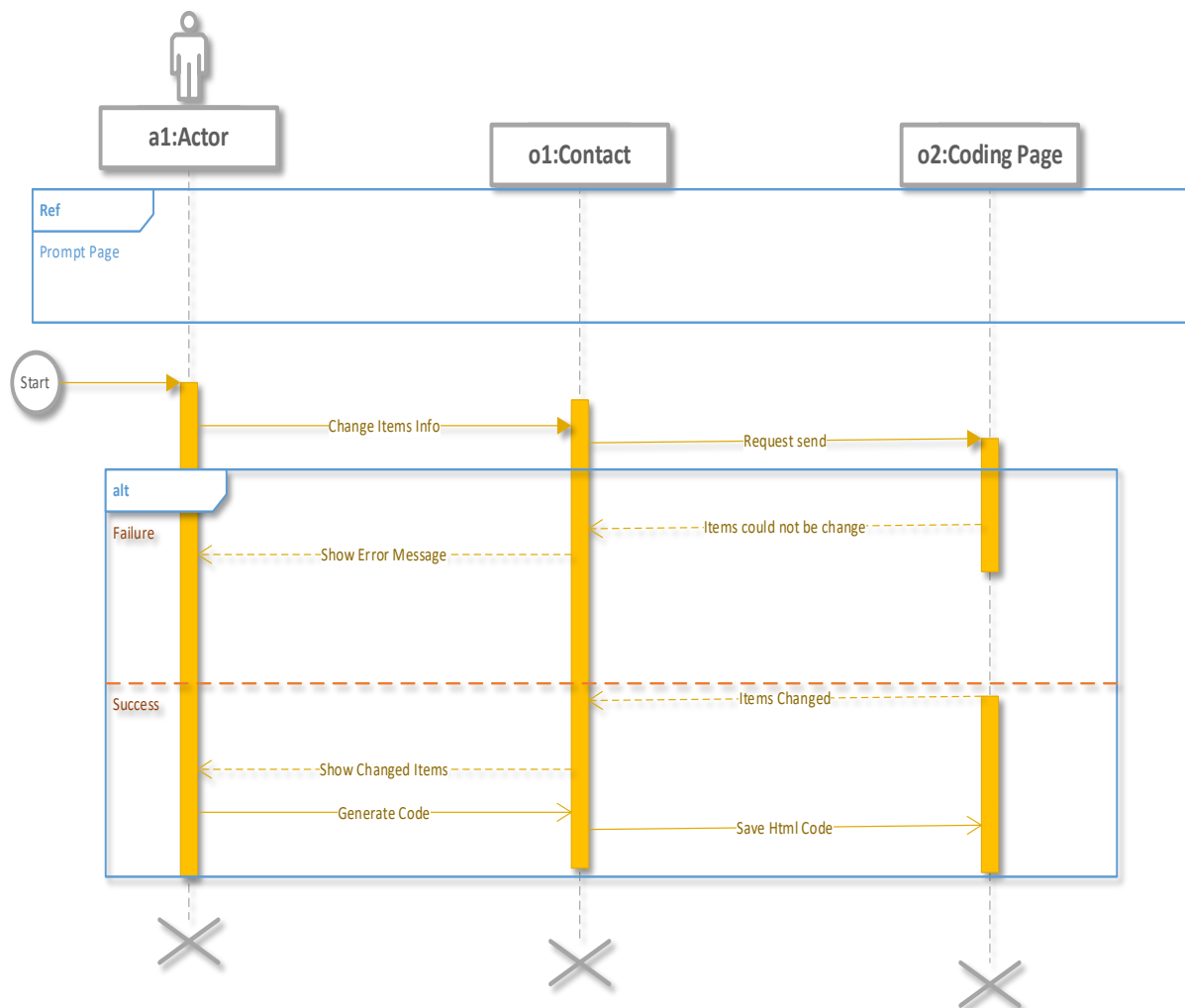


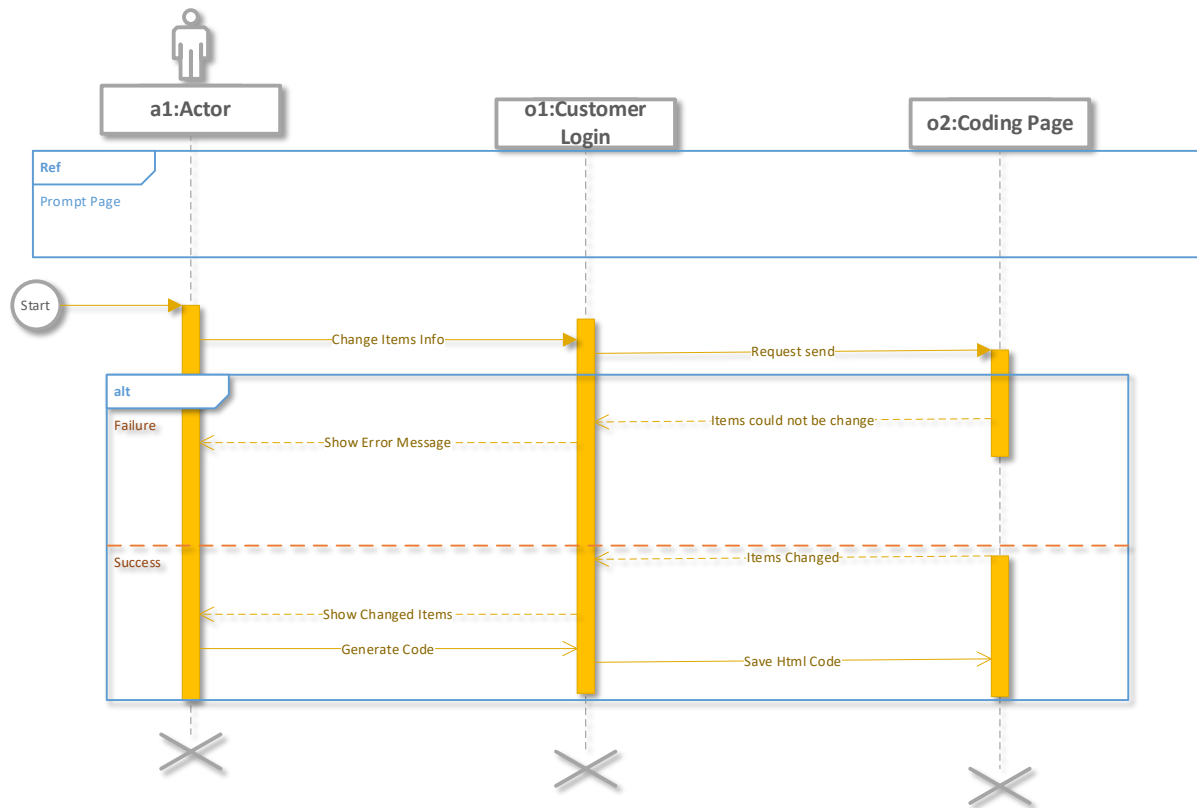


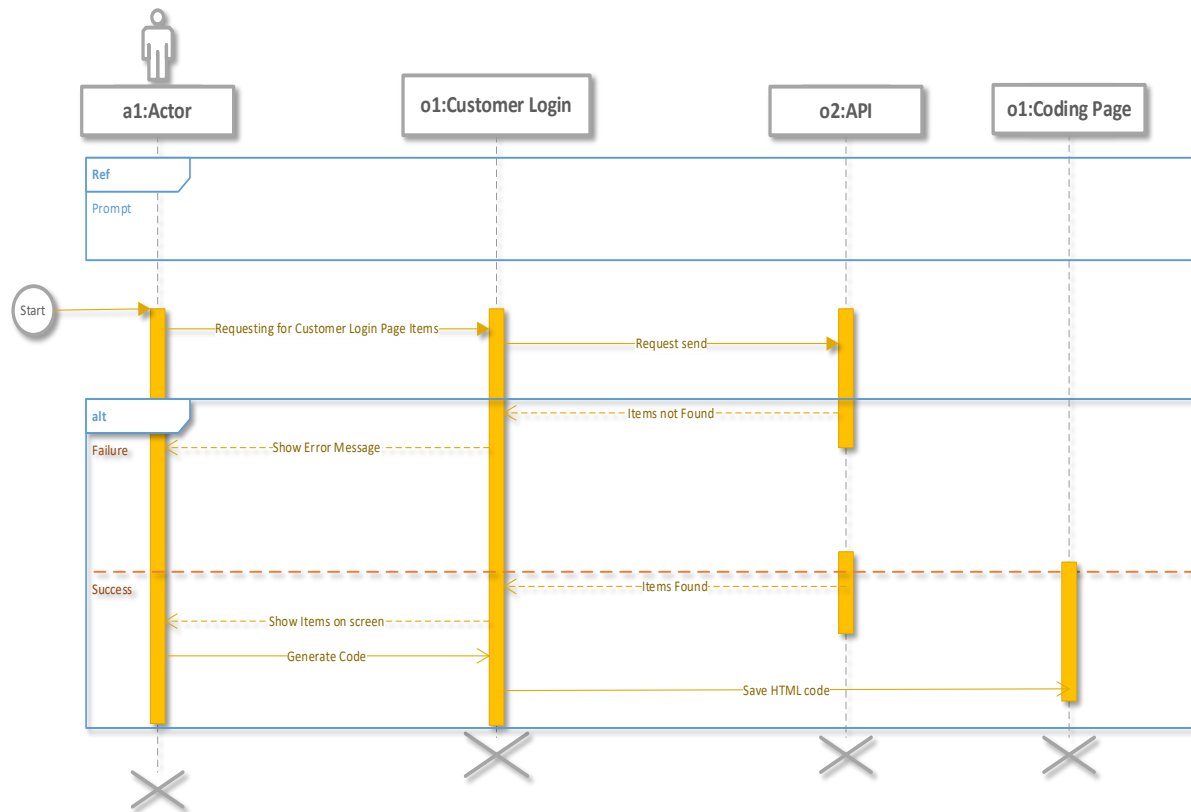


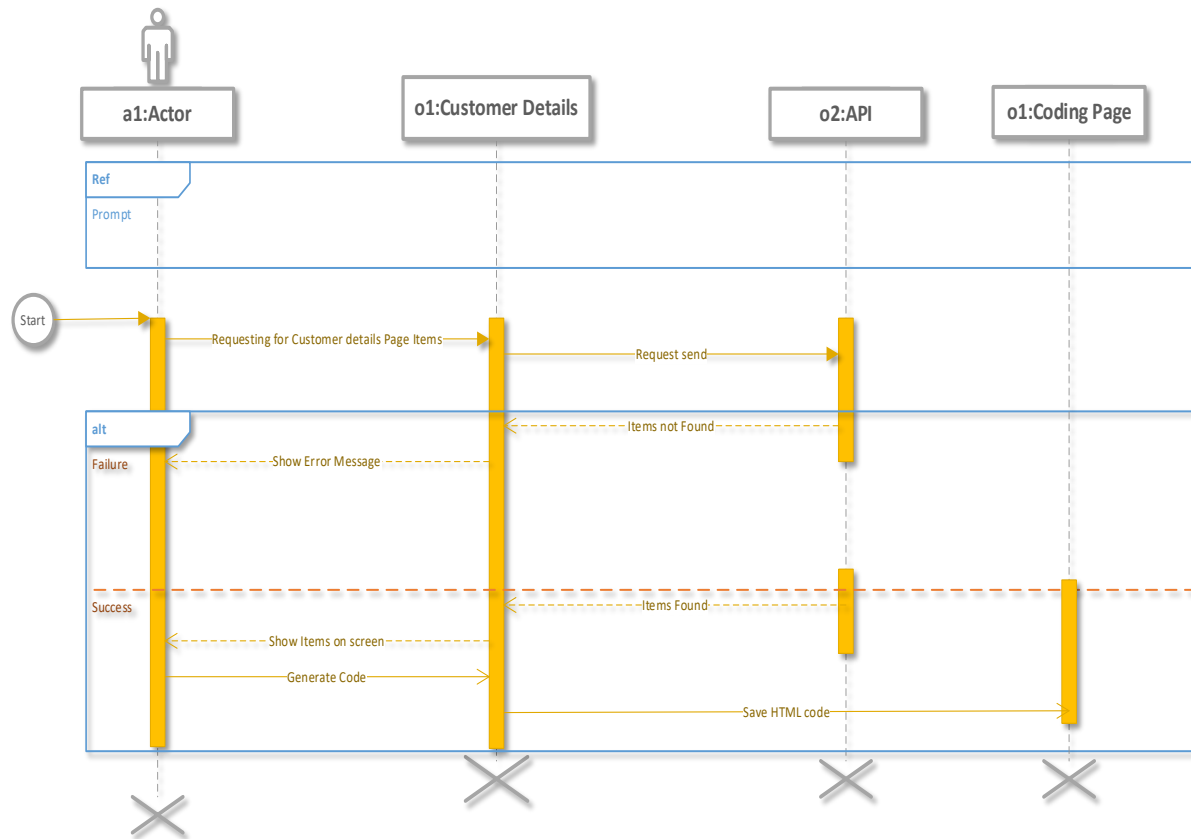


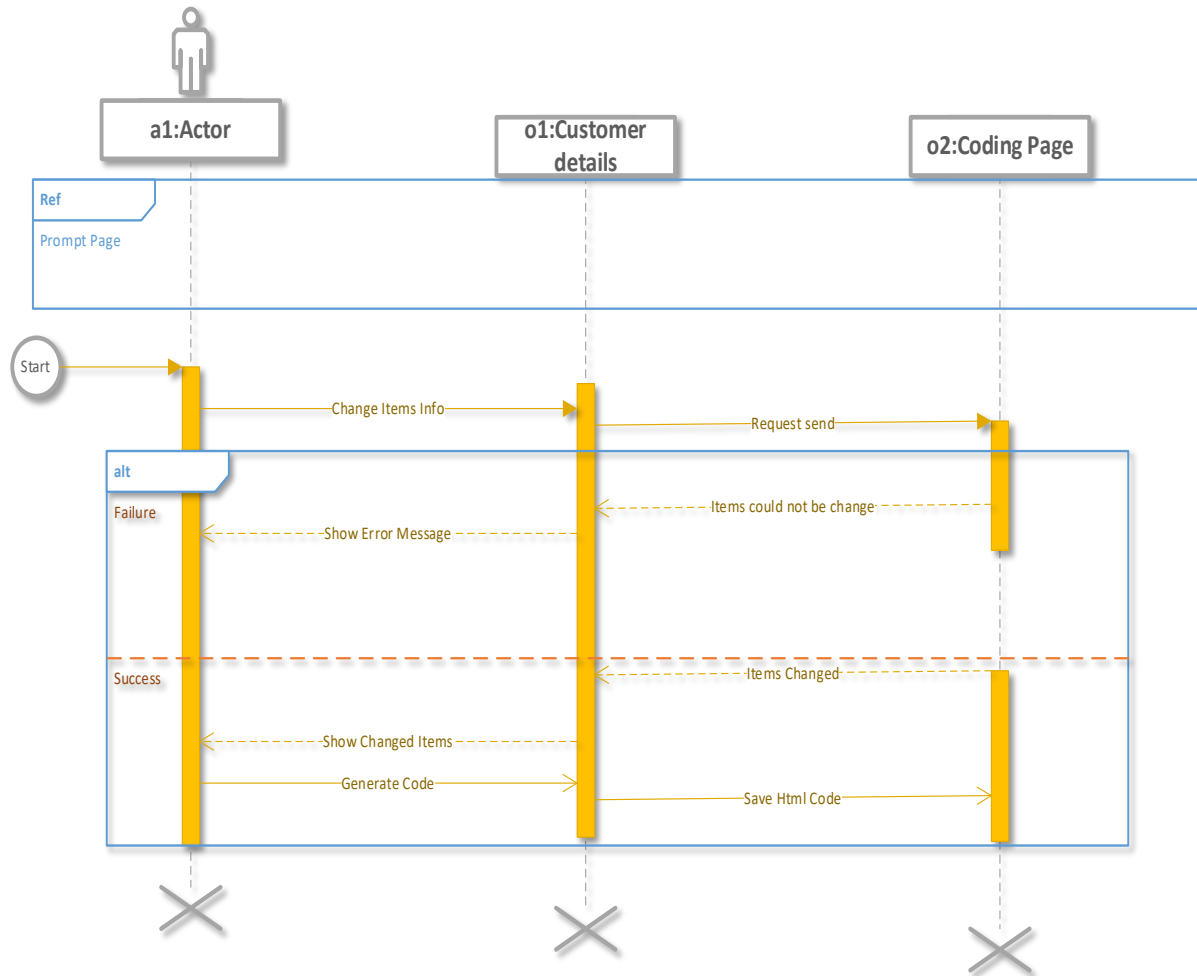


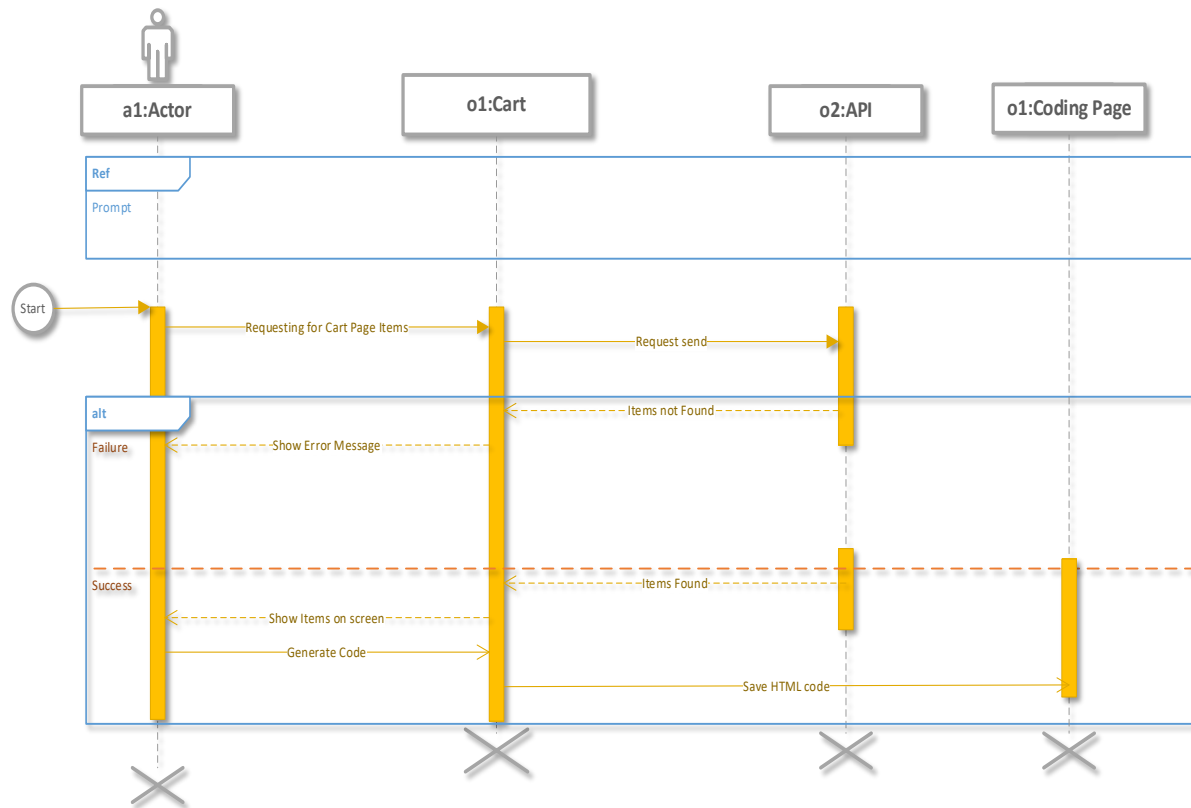


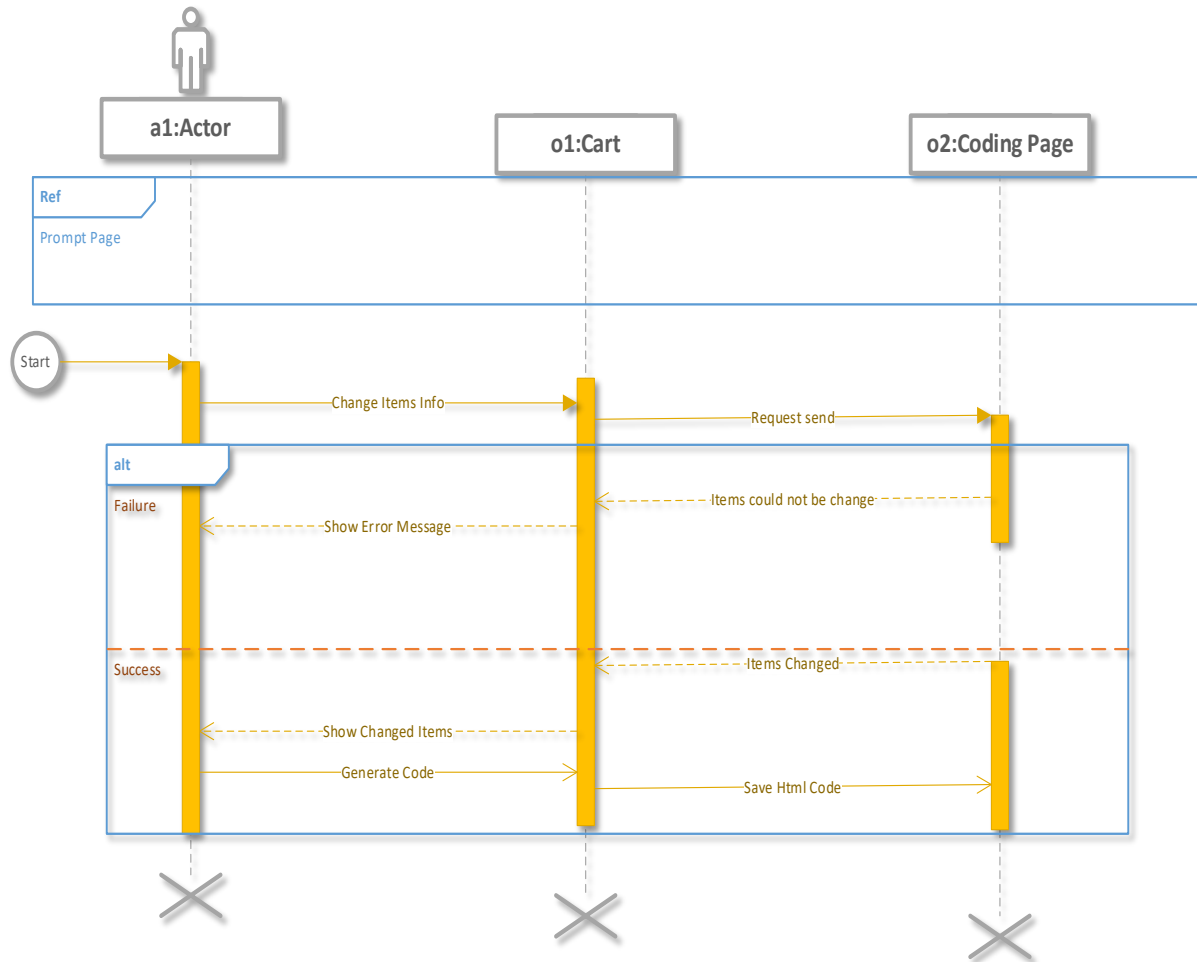


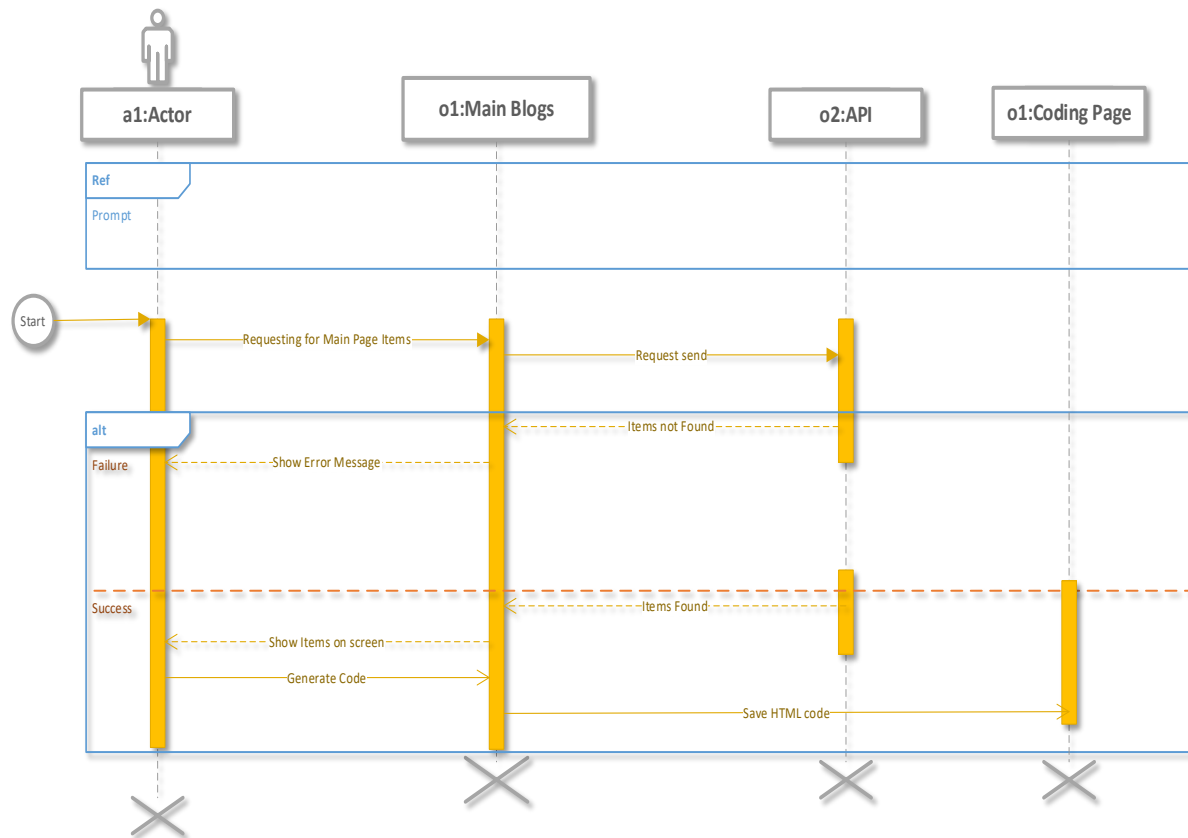


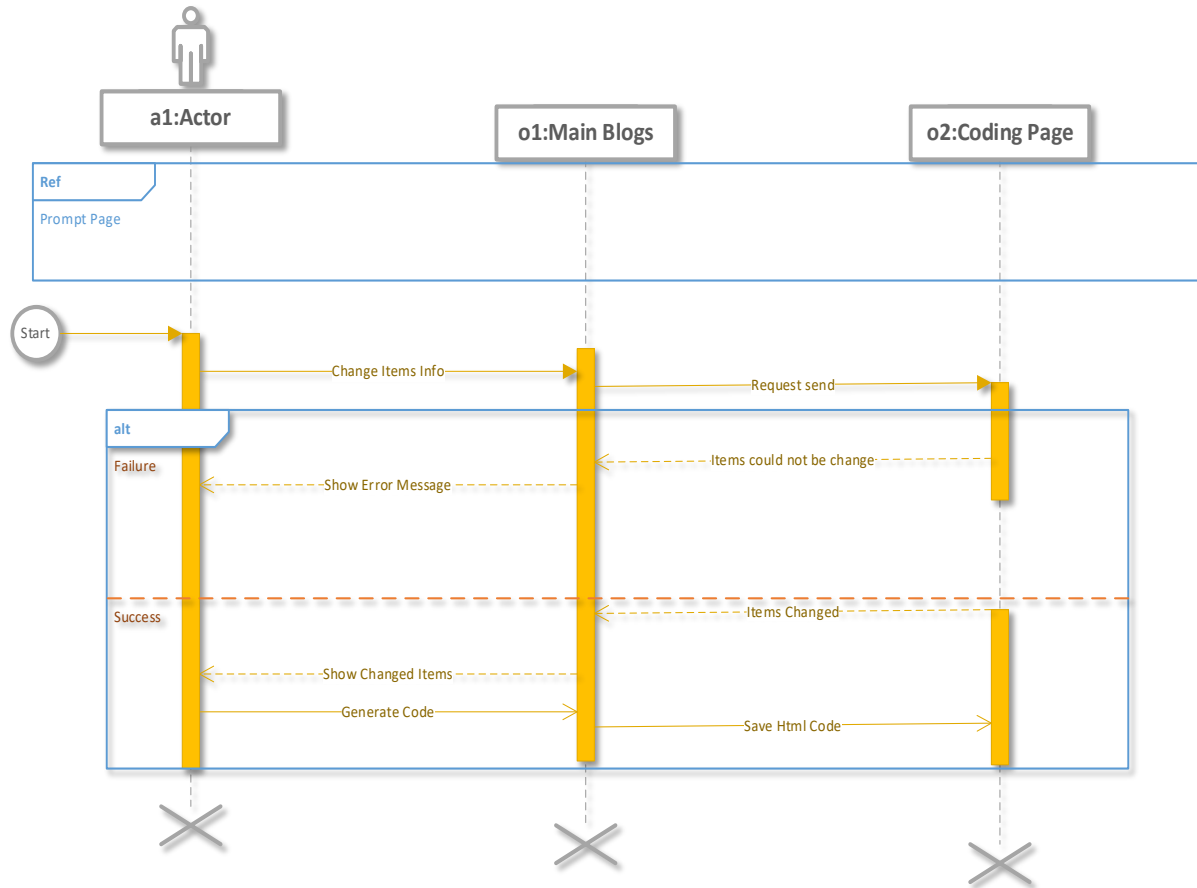


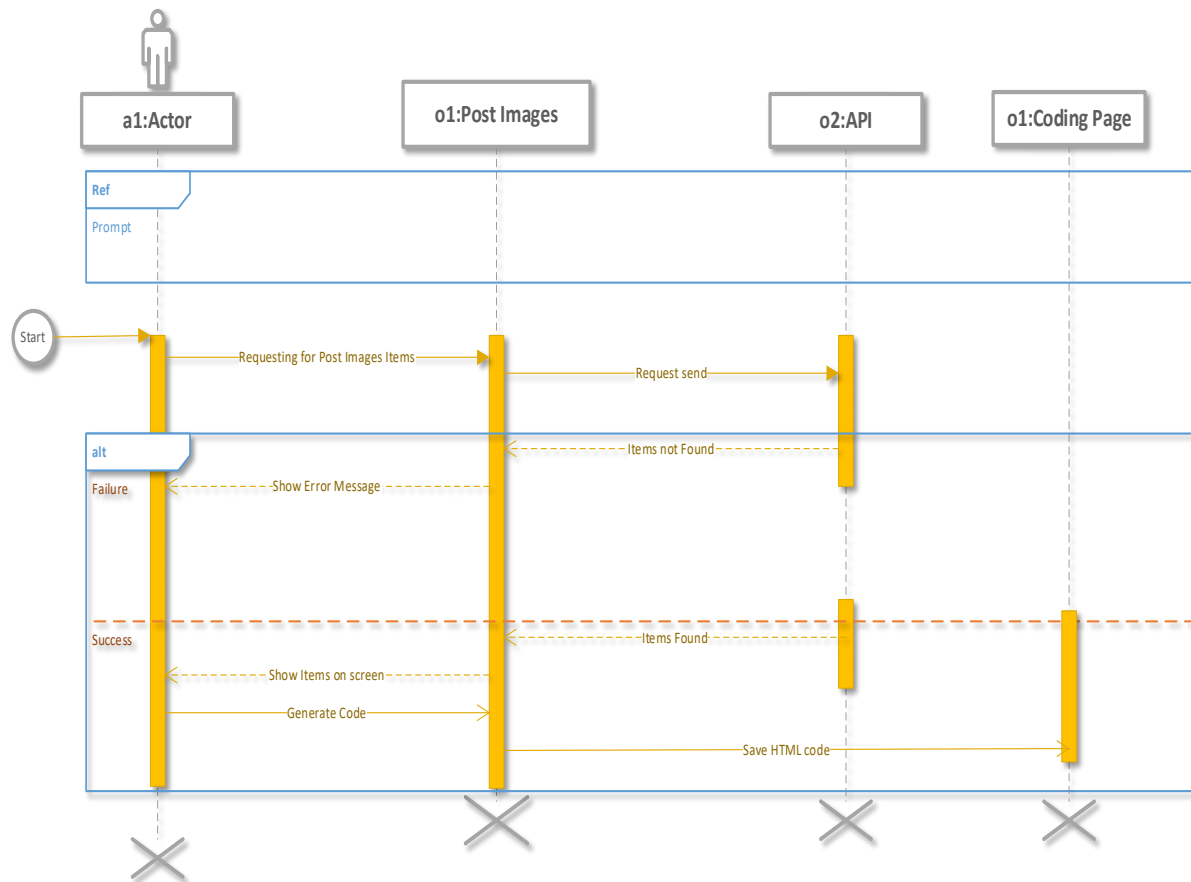


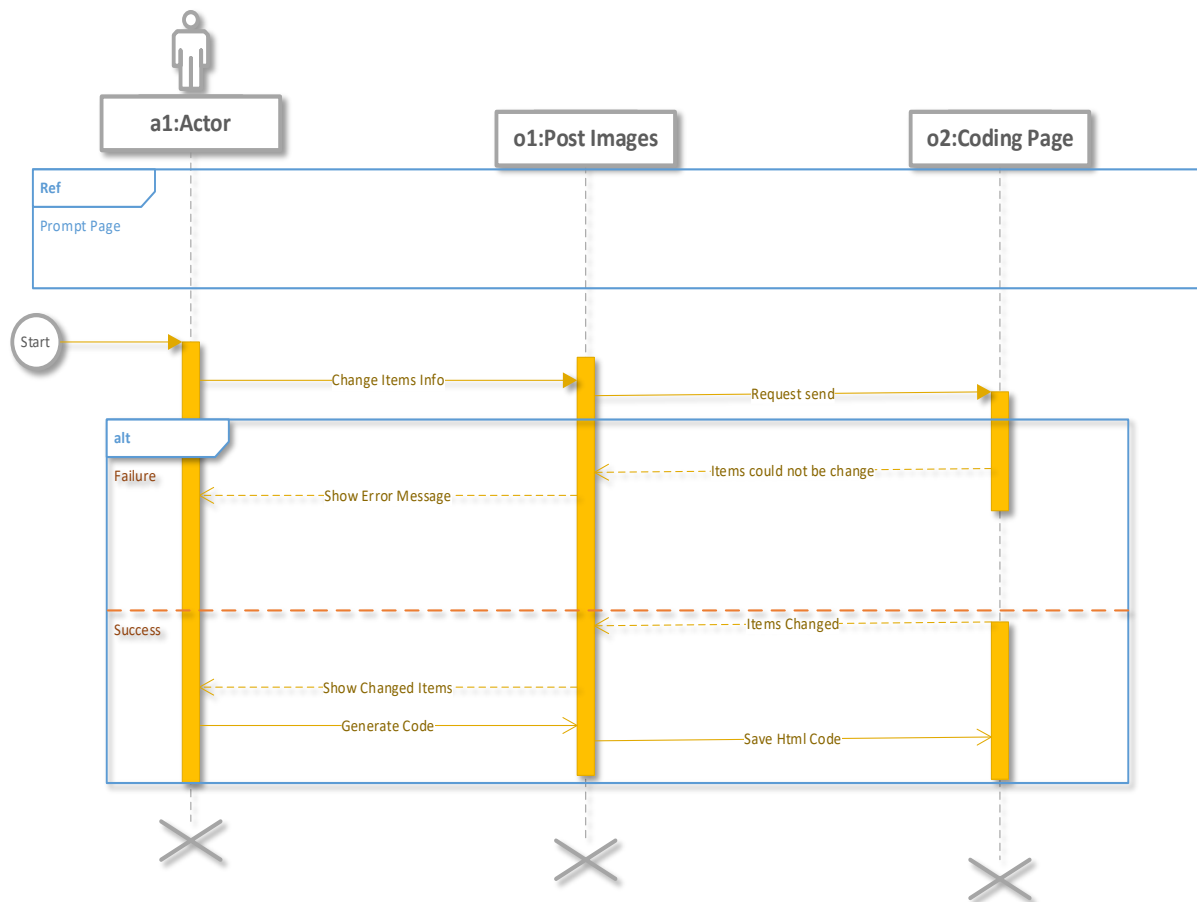


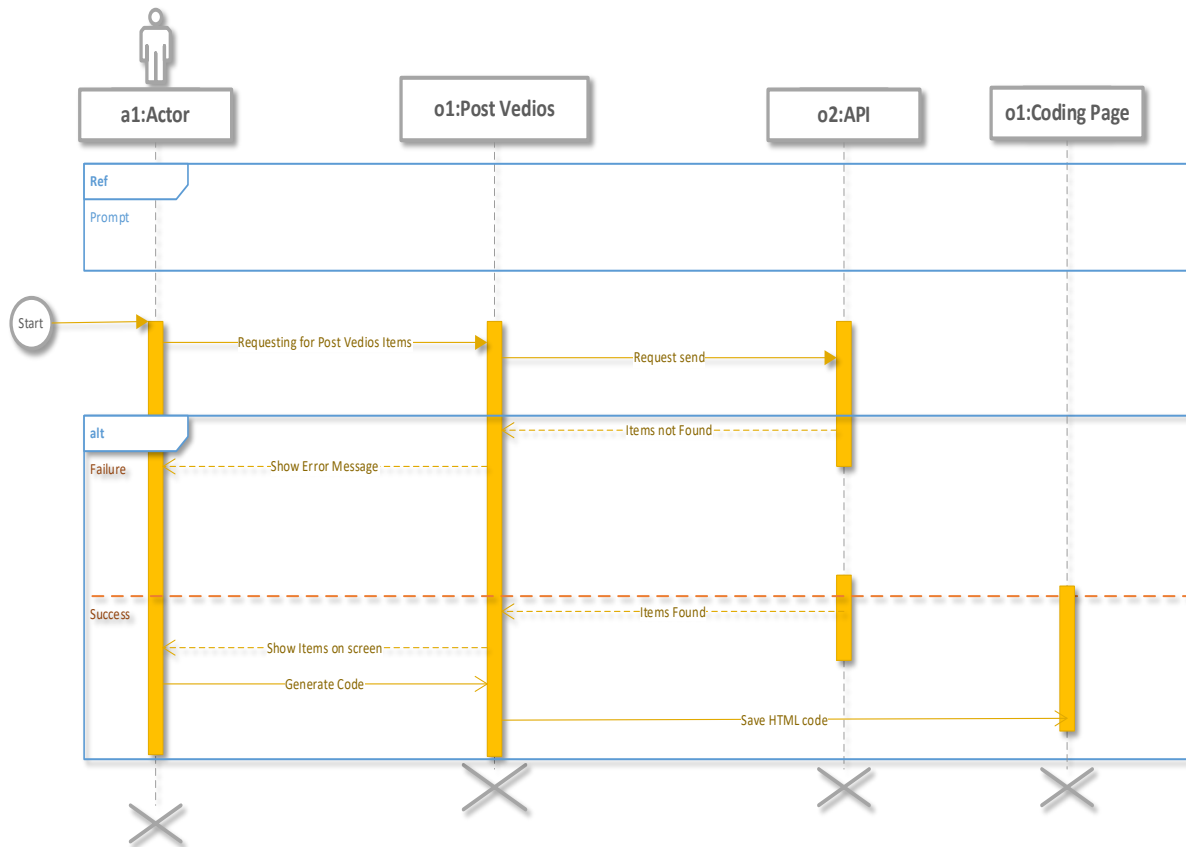


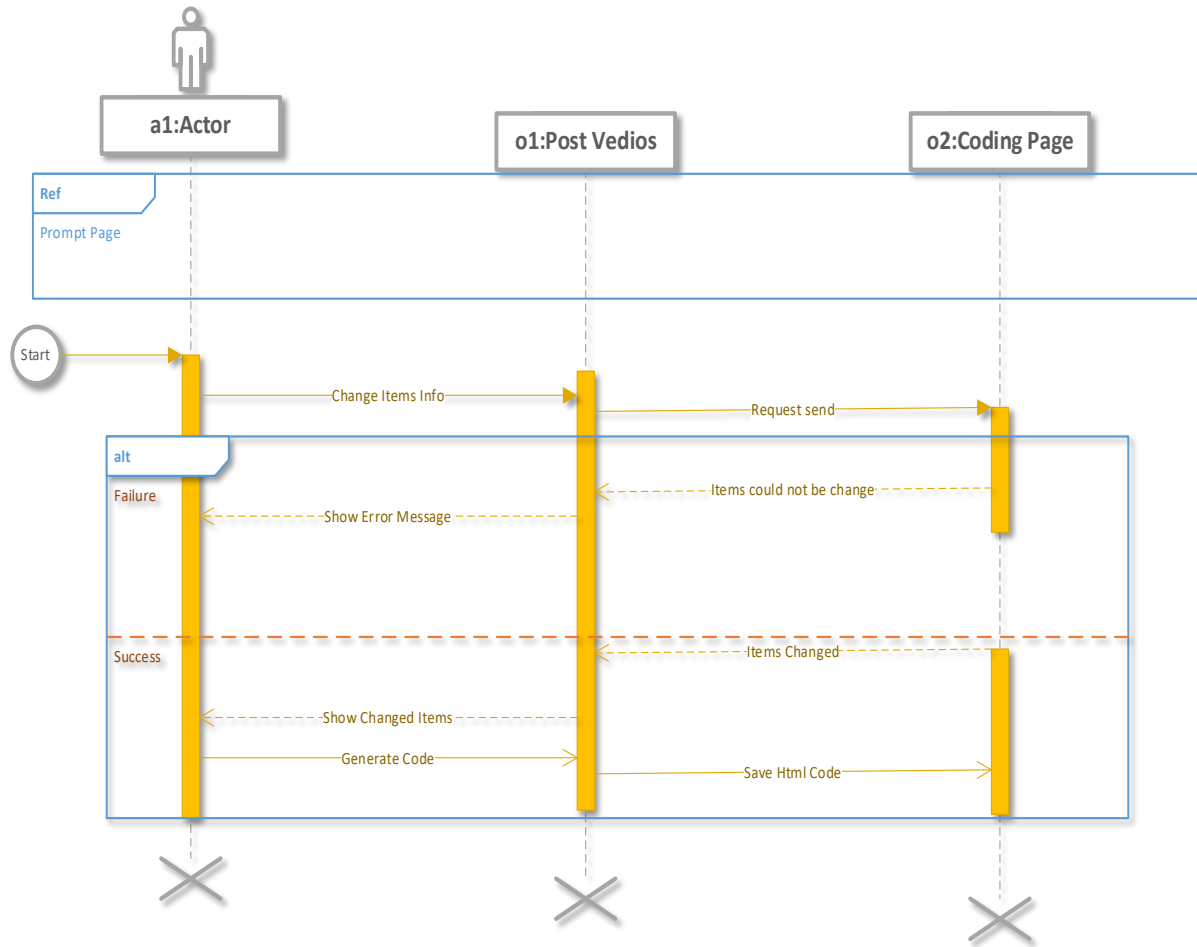


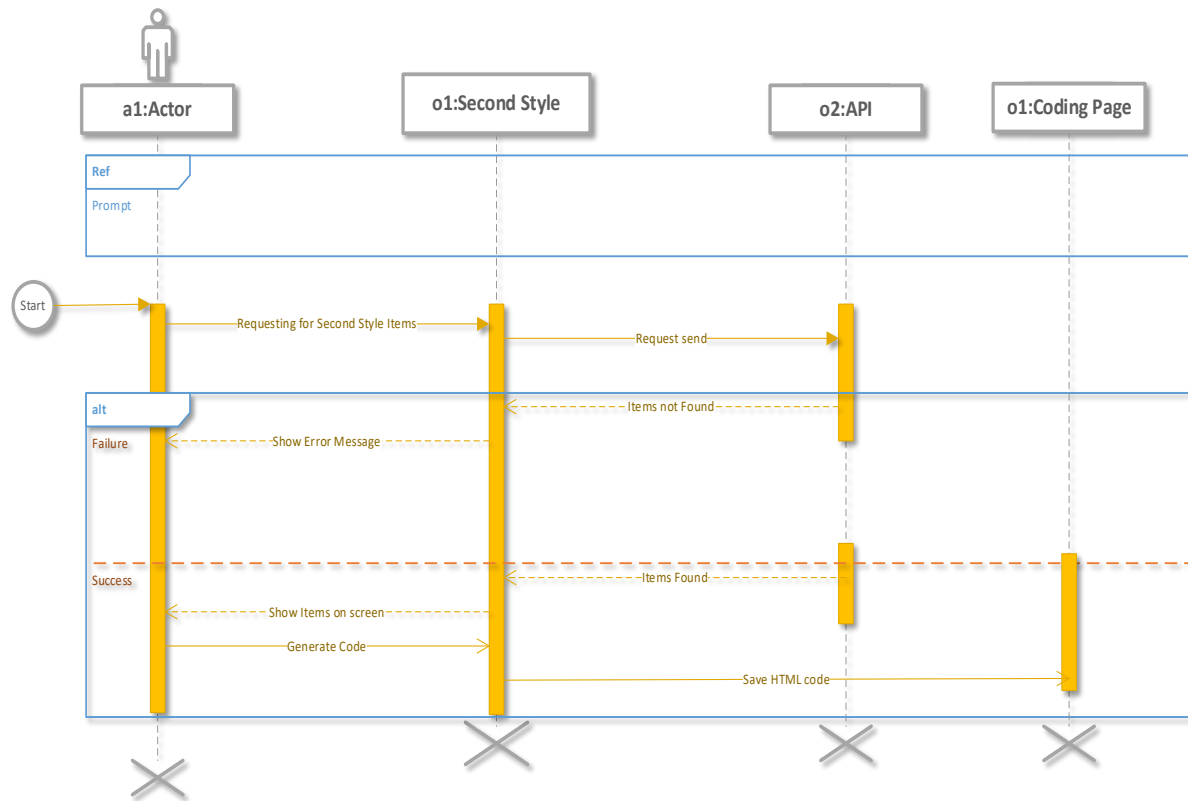


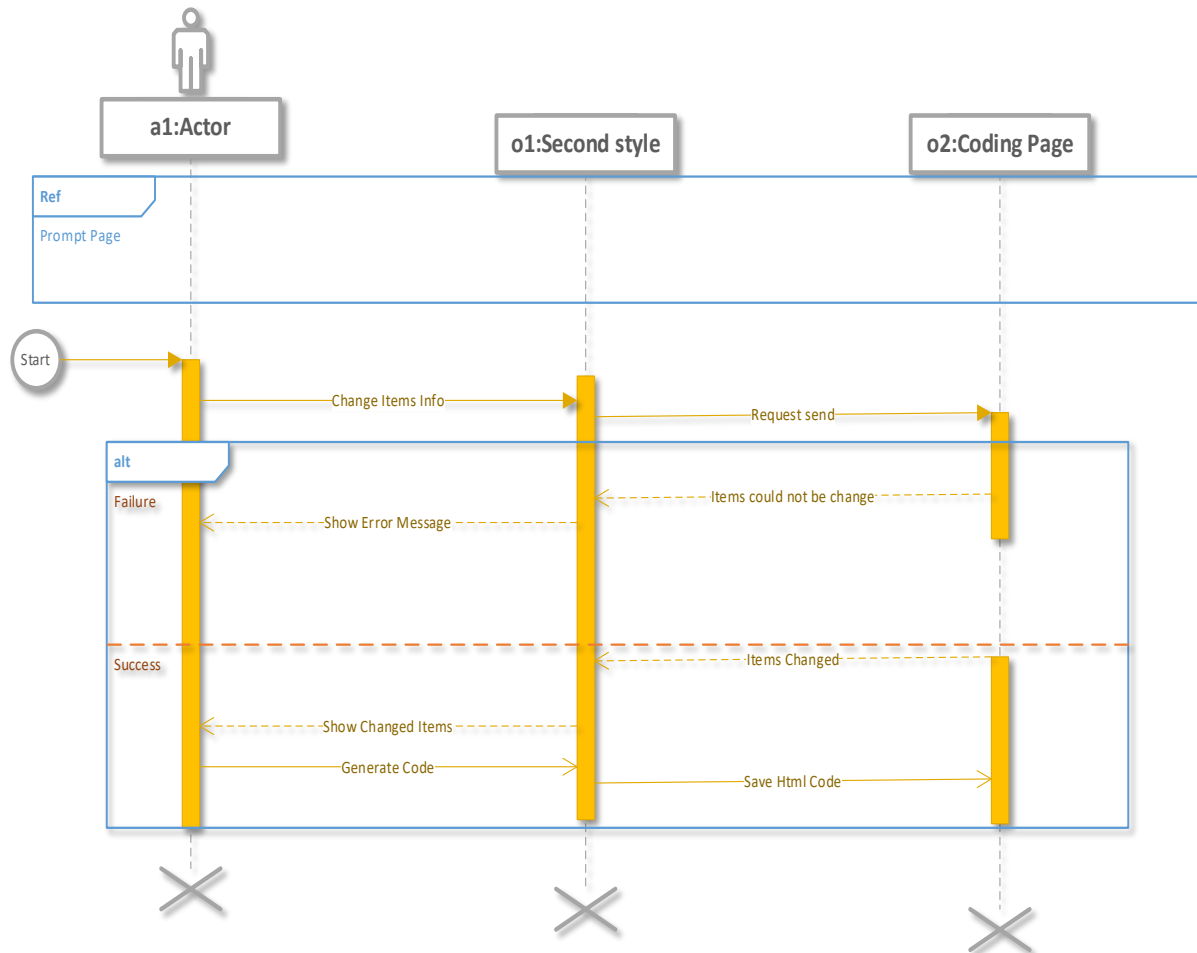


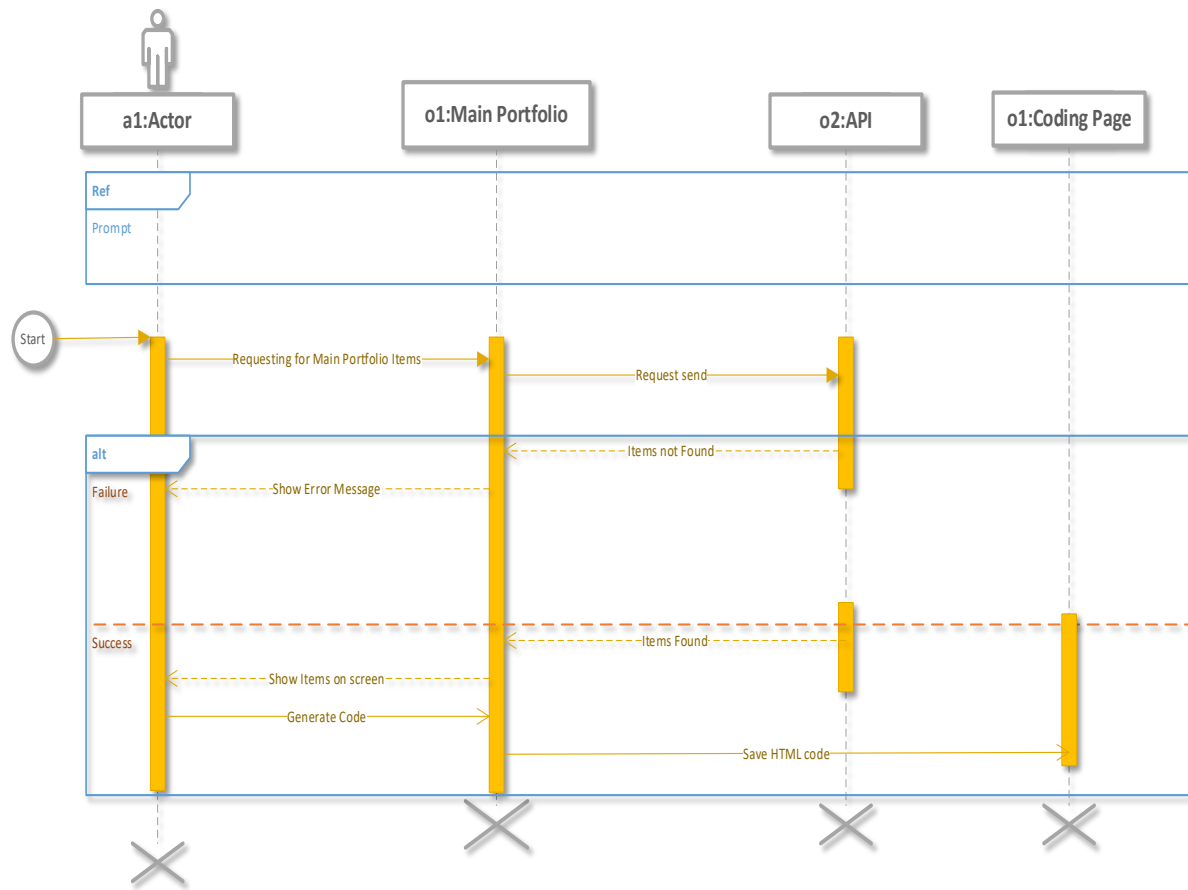


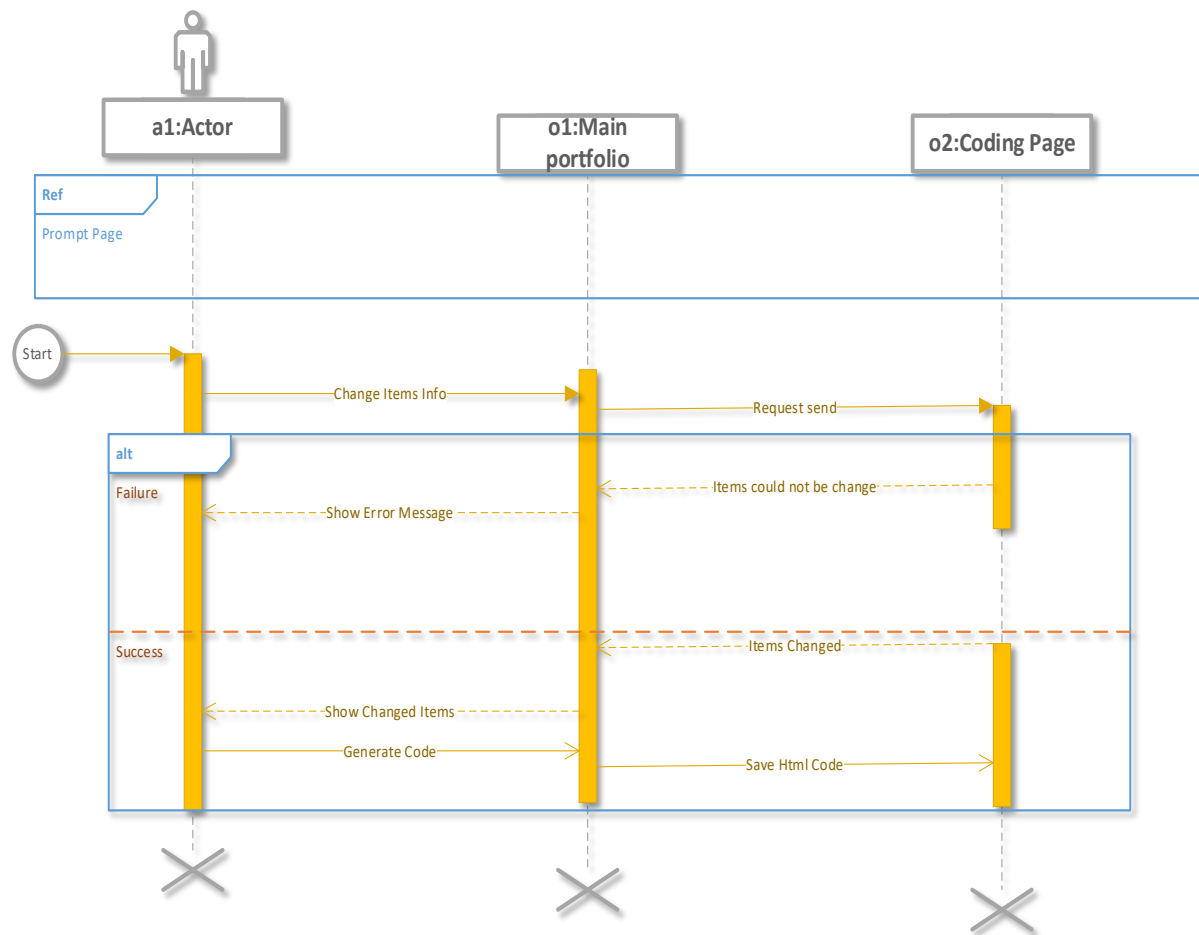


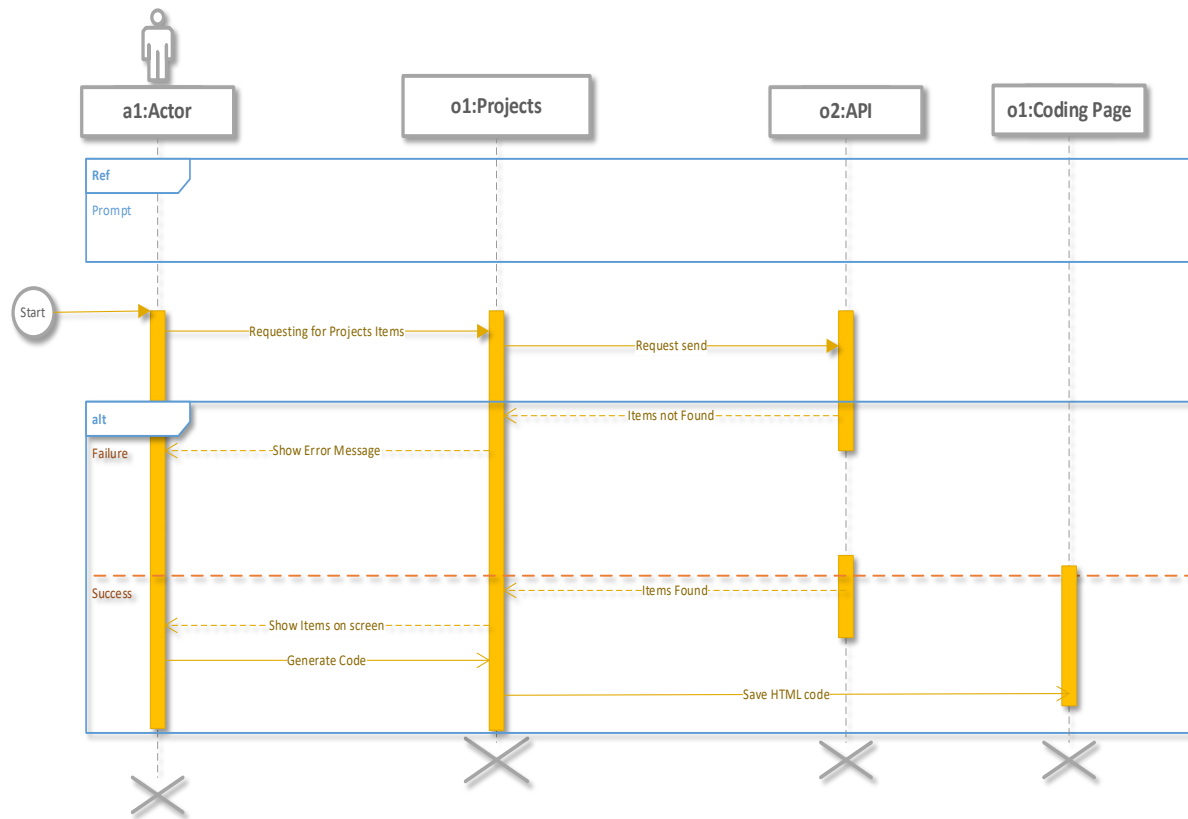


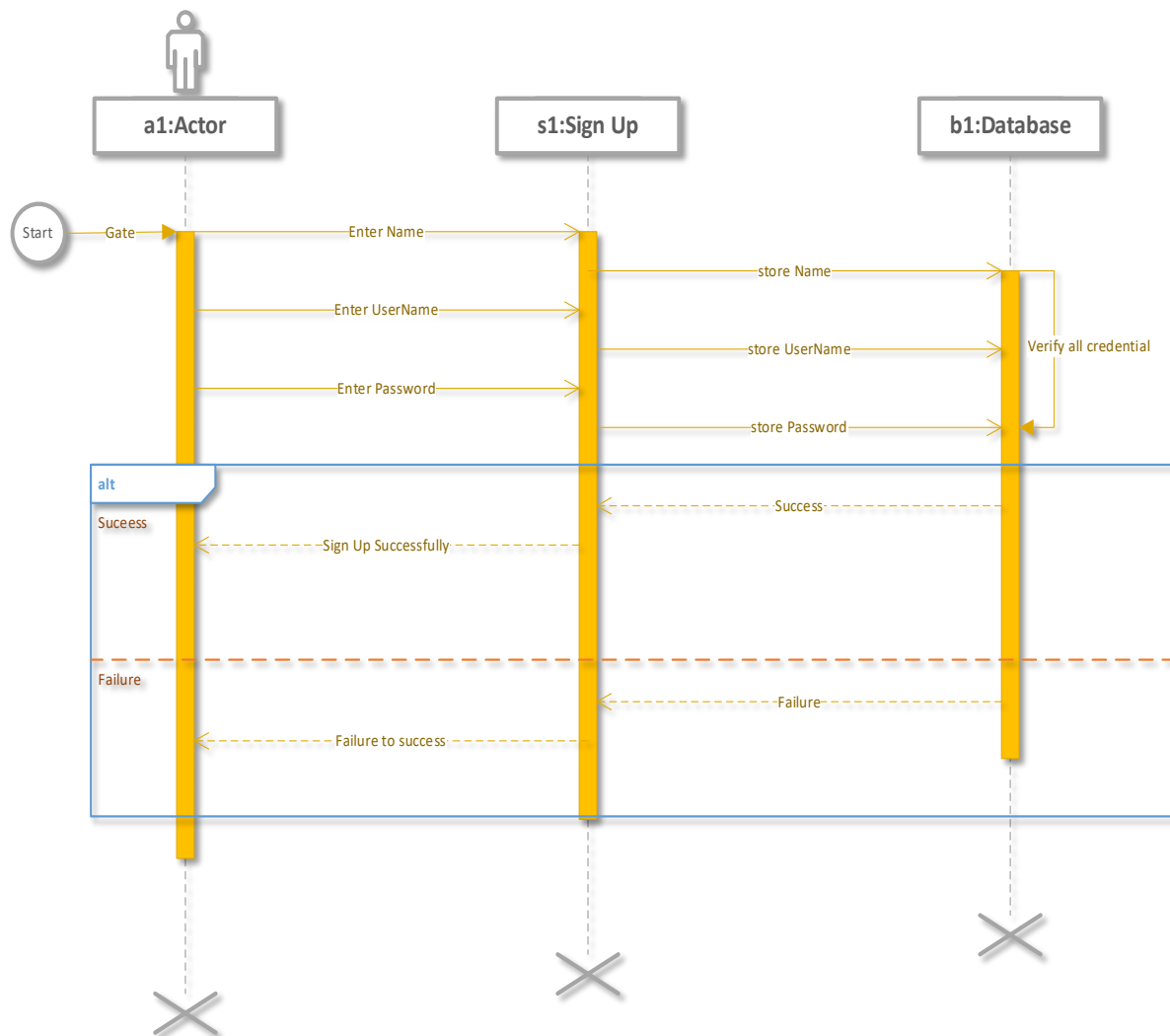


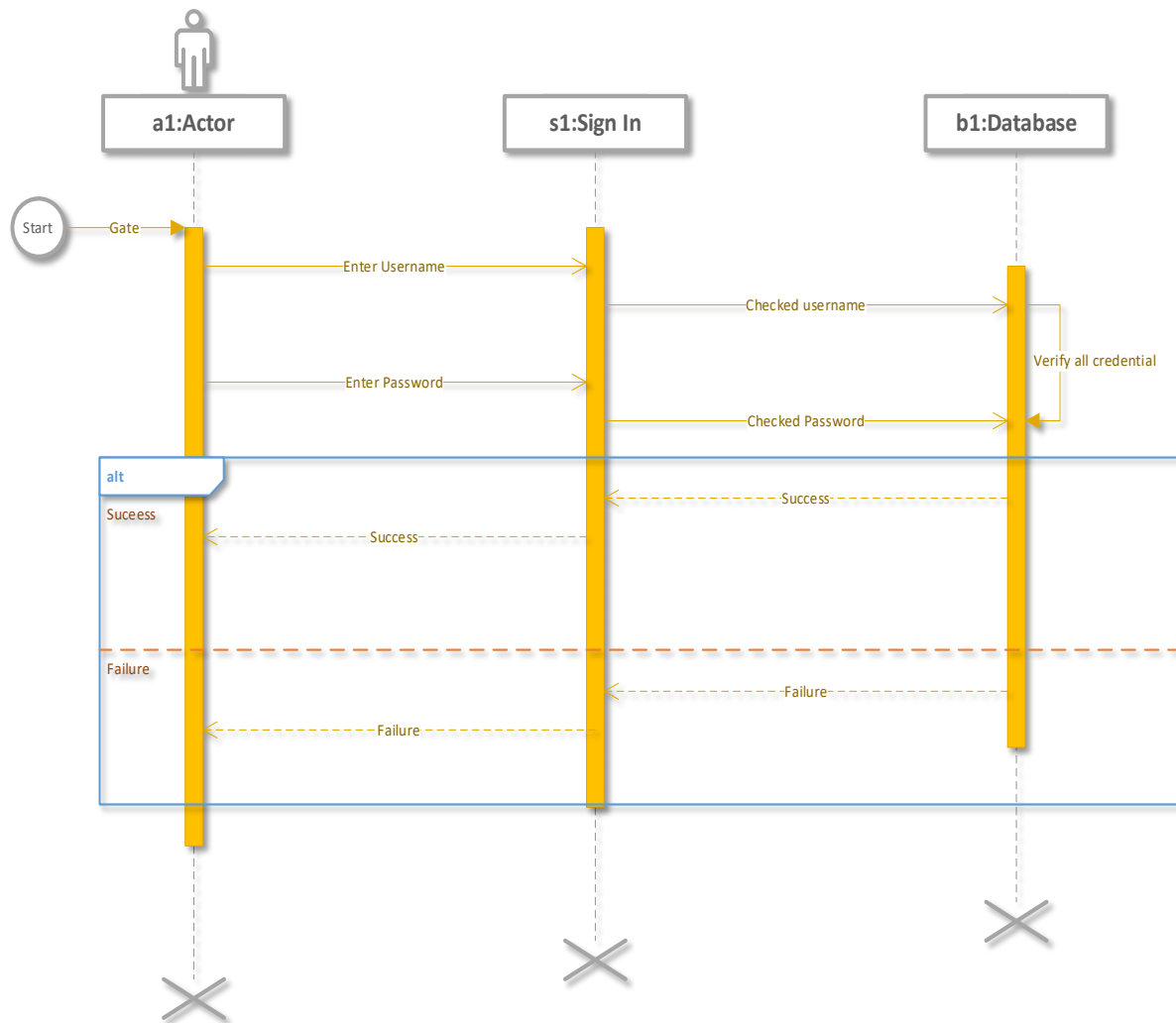


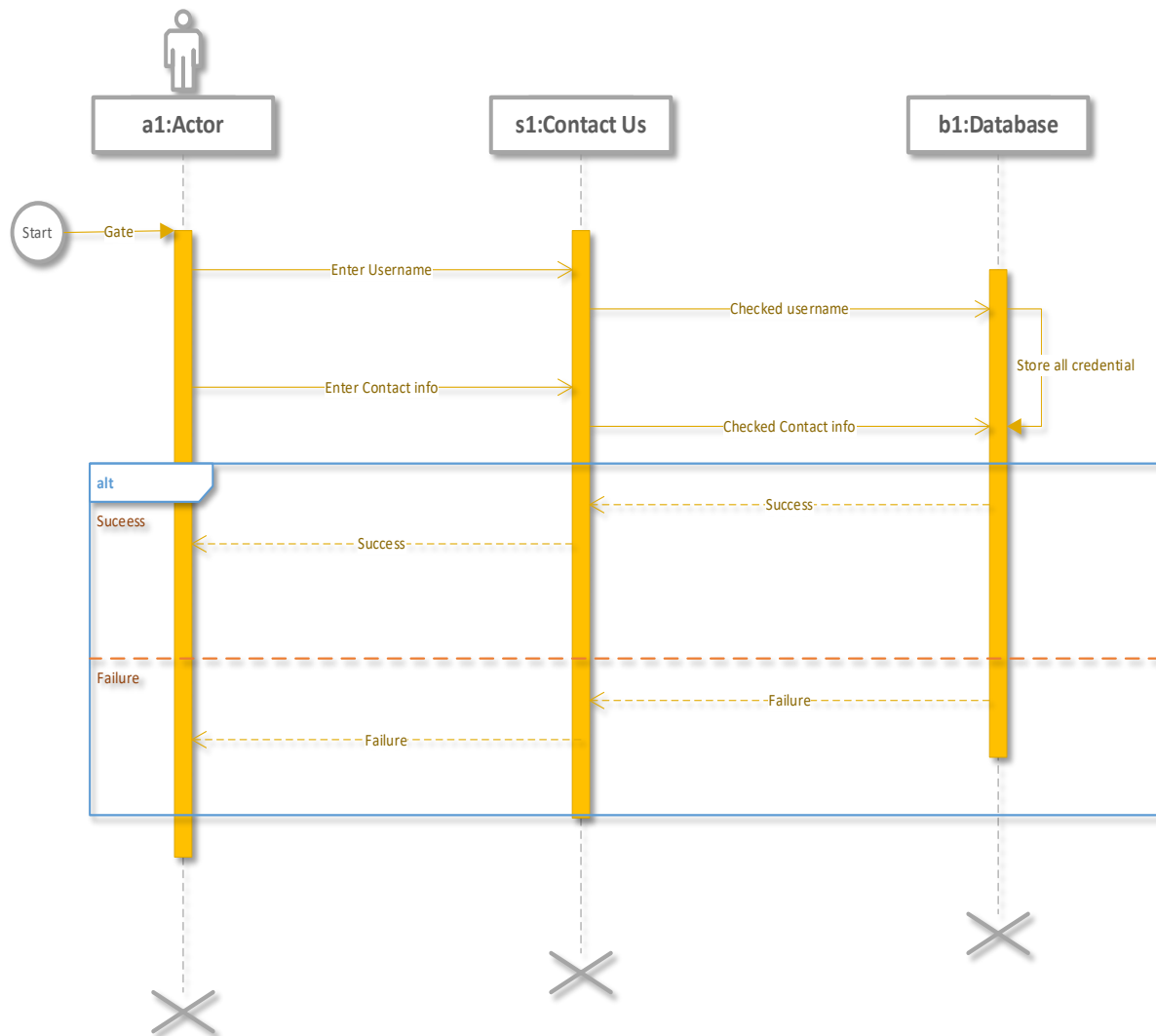


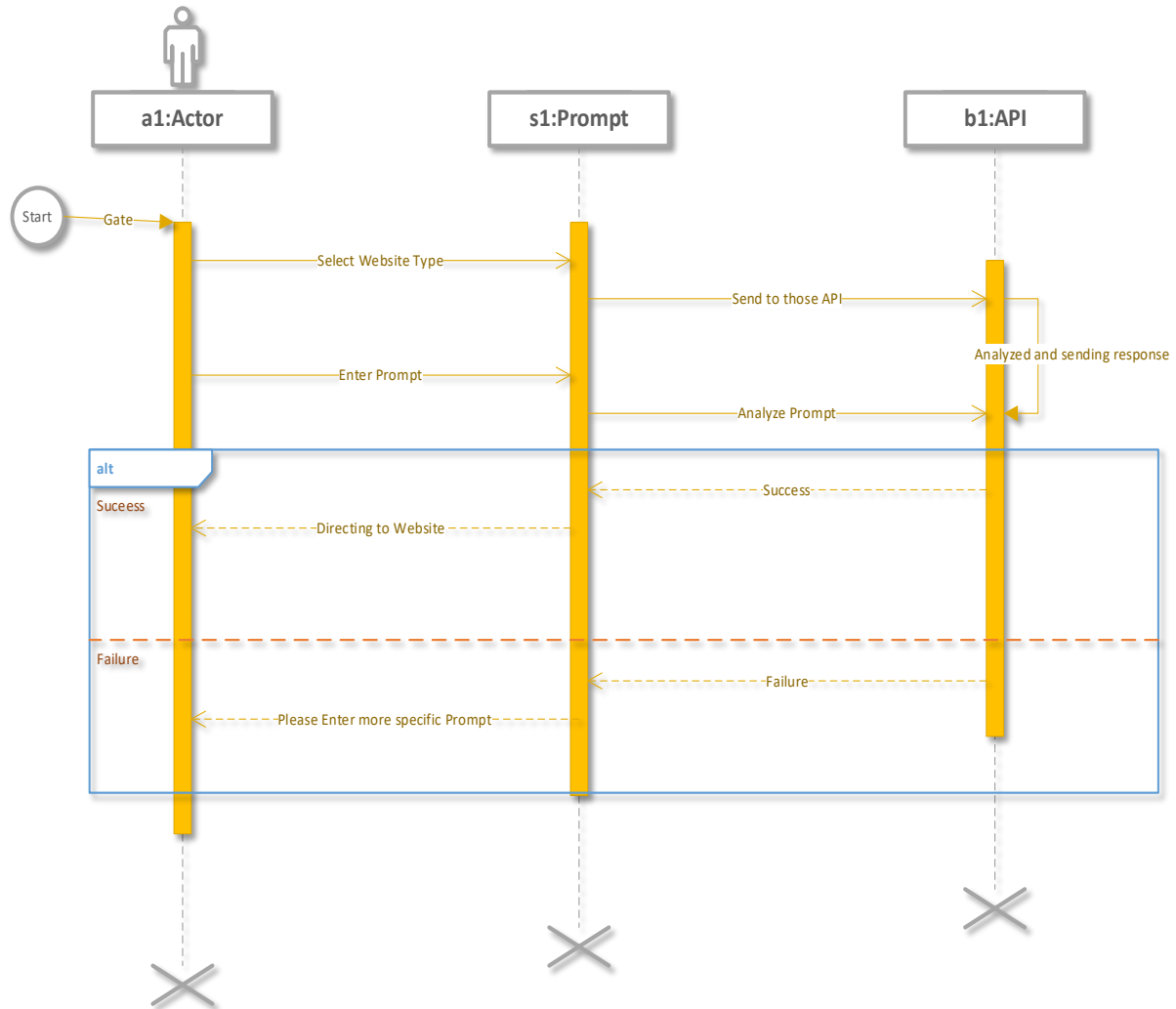












2.1.5 Collaboration Diagram

2.1.5.1 Admin End

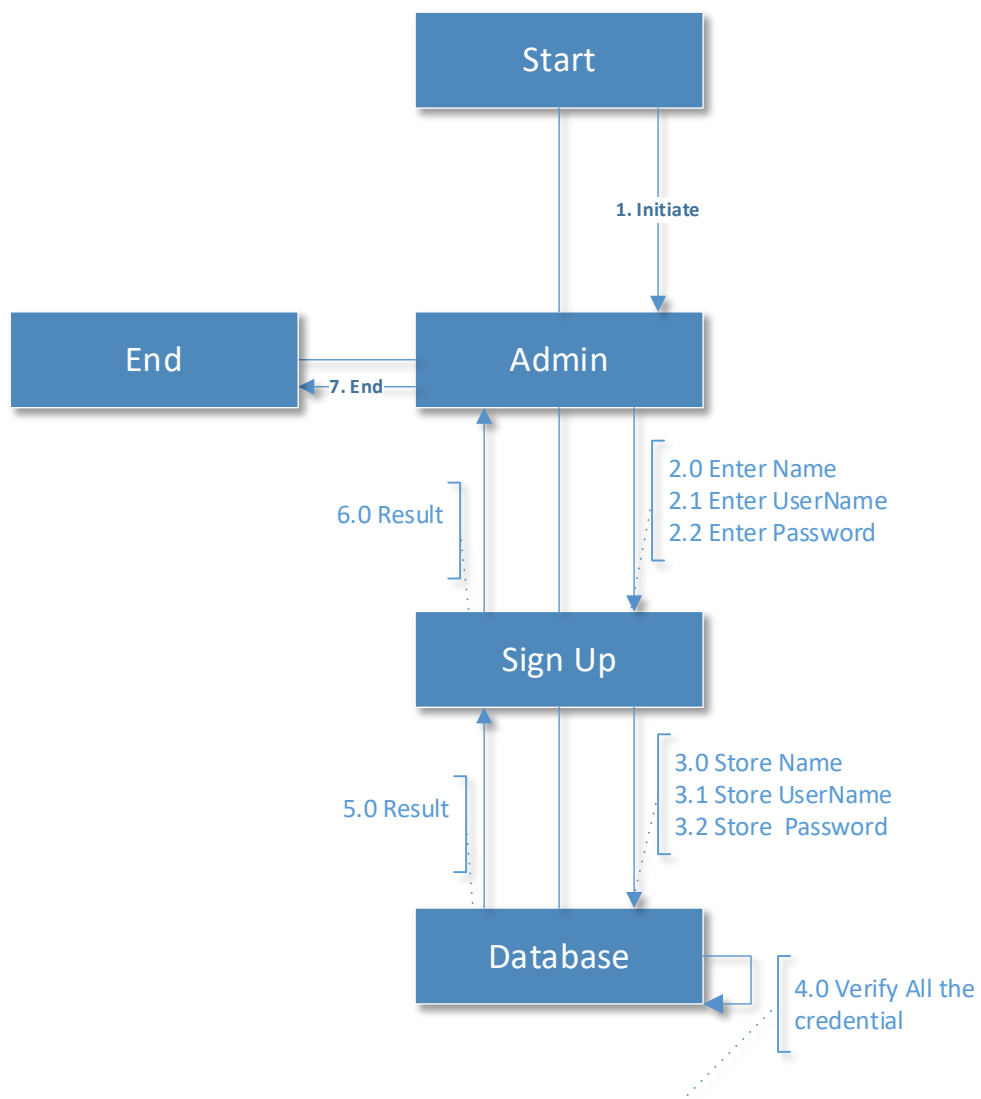
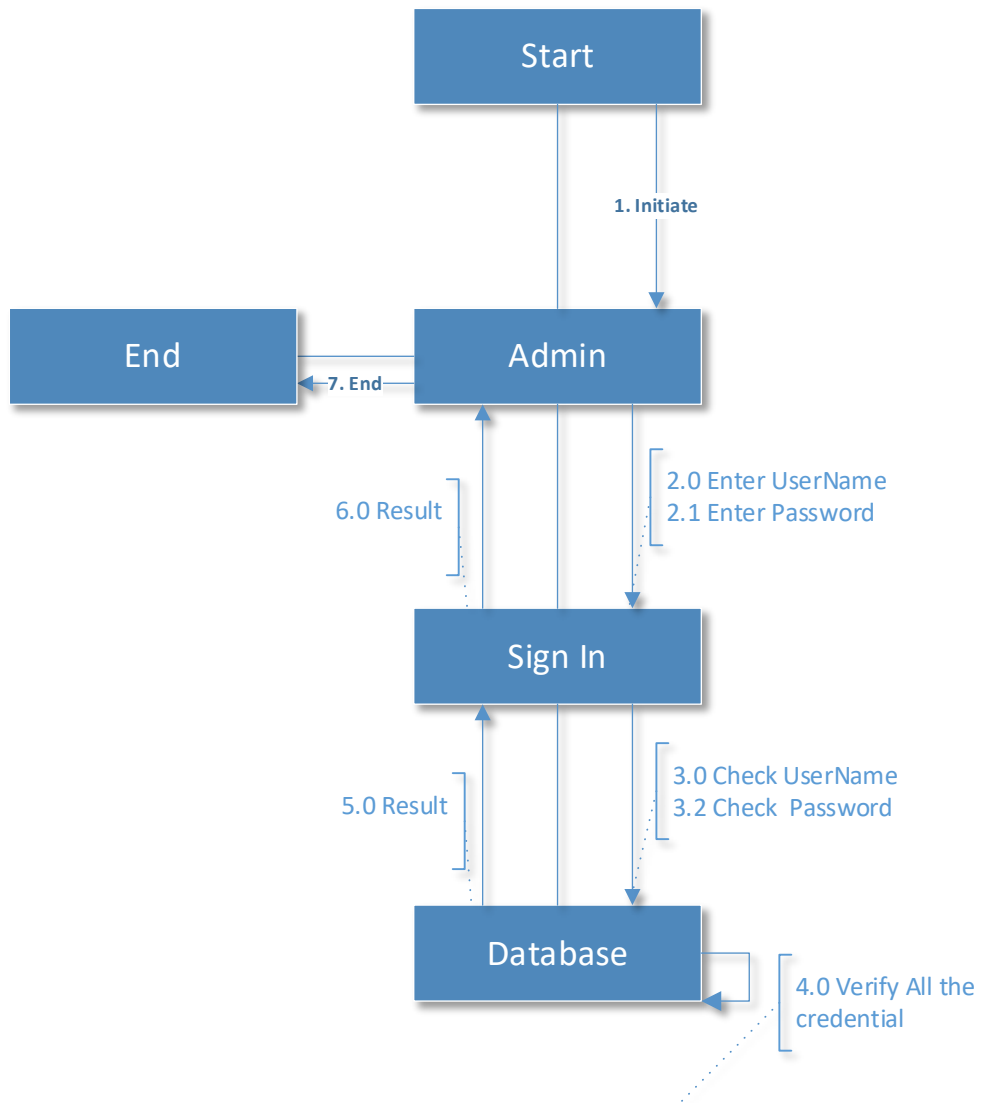
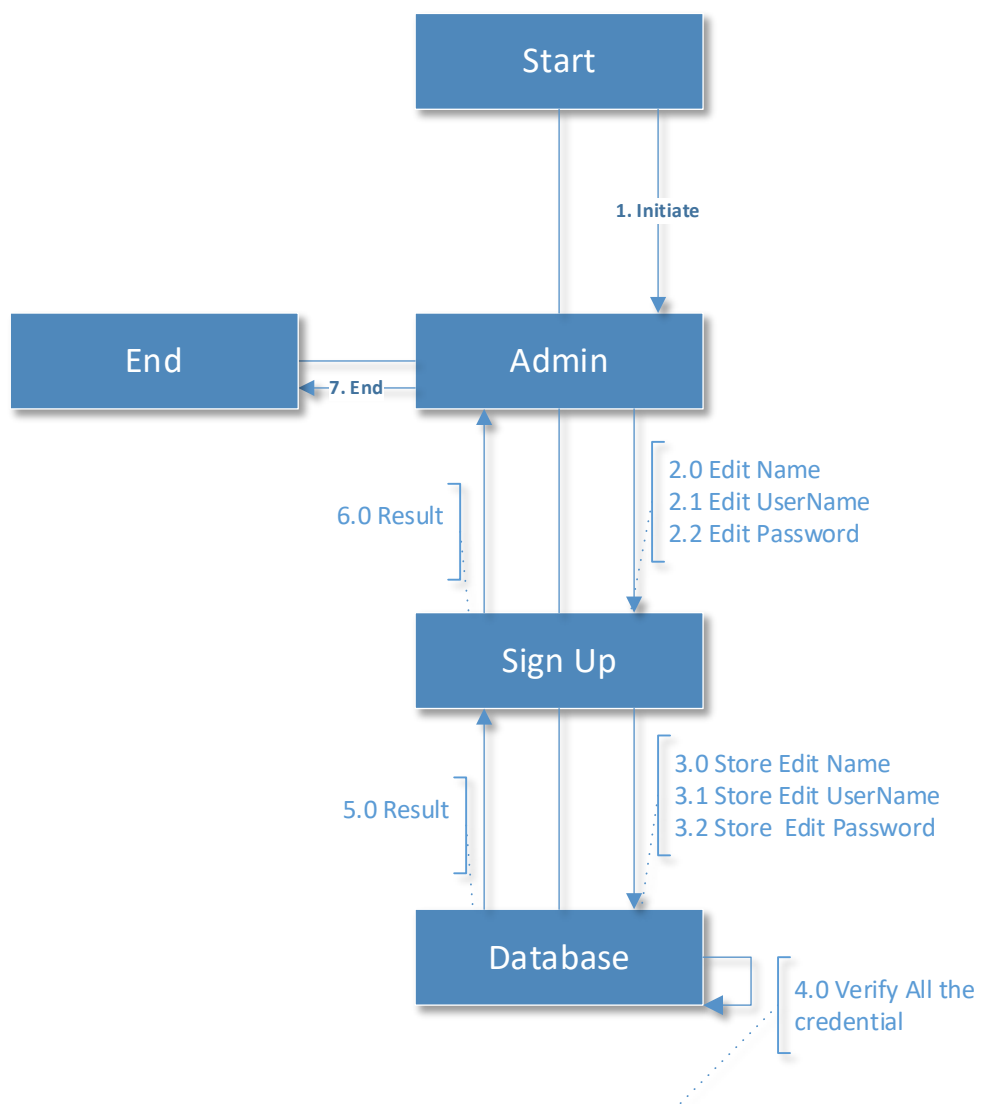
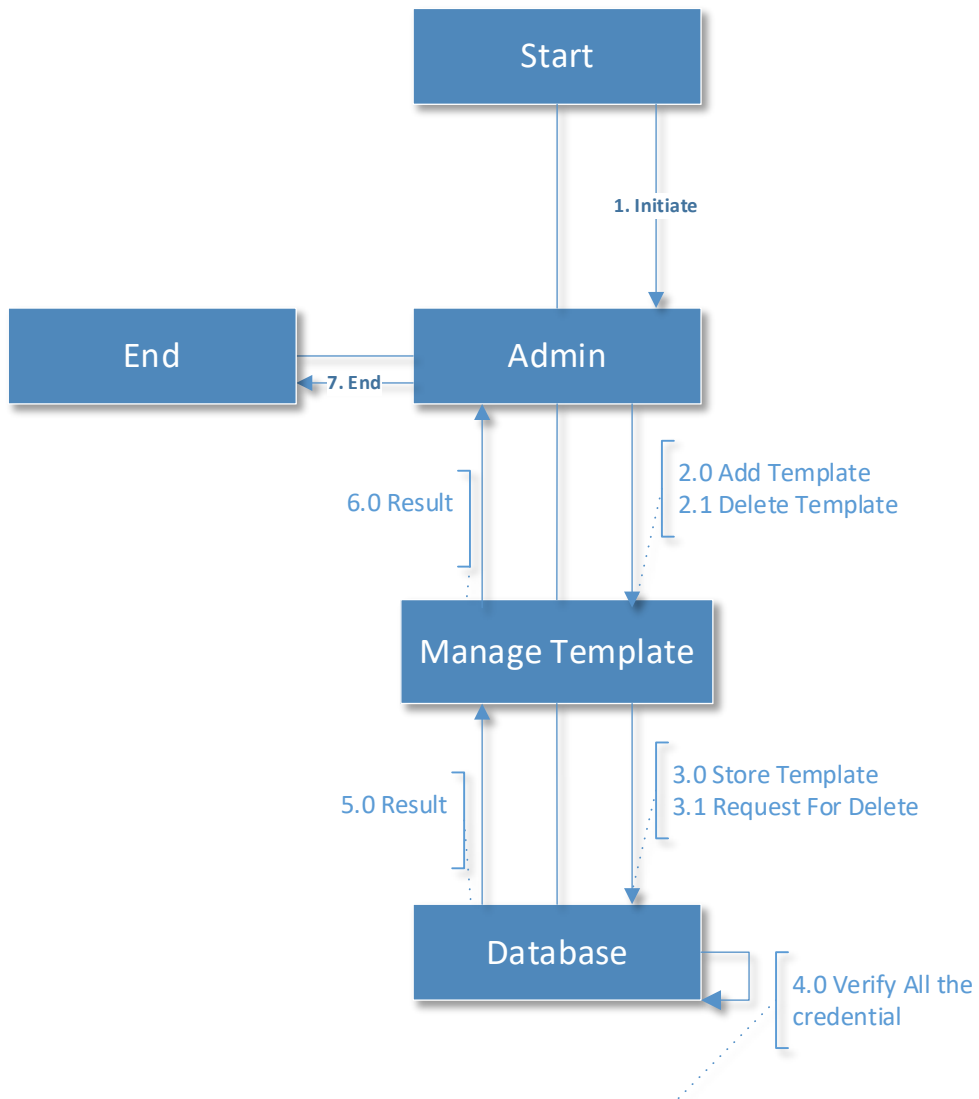


Figure 27: Admin End Collaboration







2.1.5.2 User End

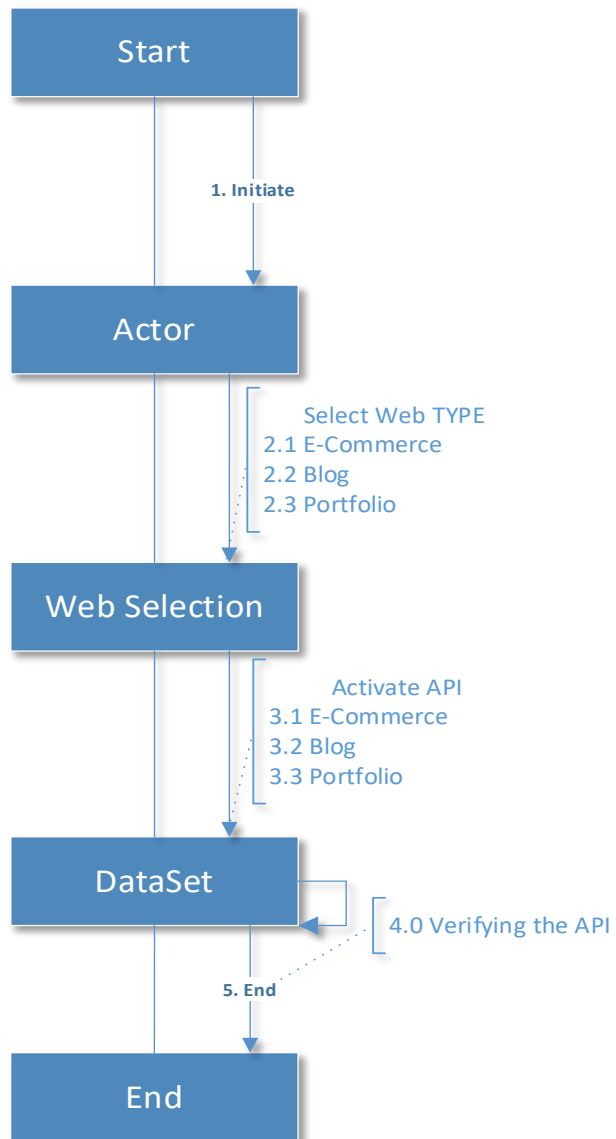
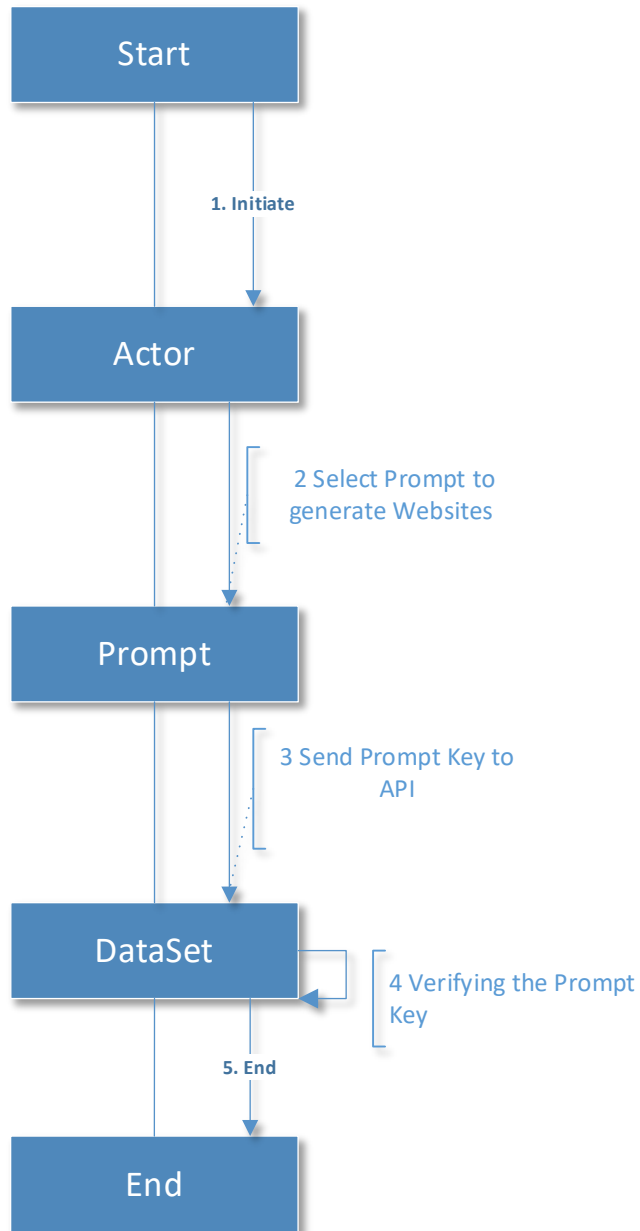
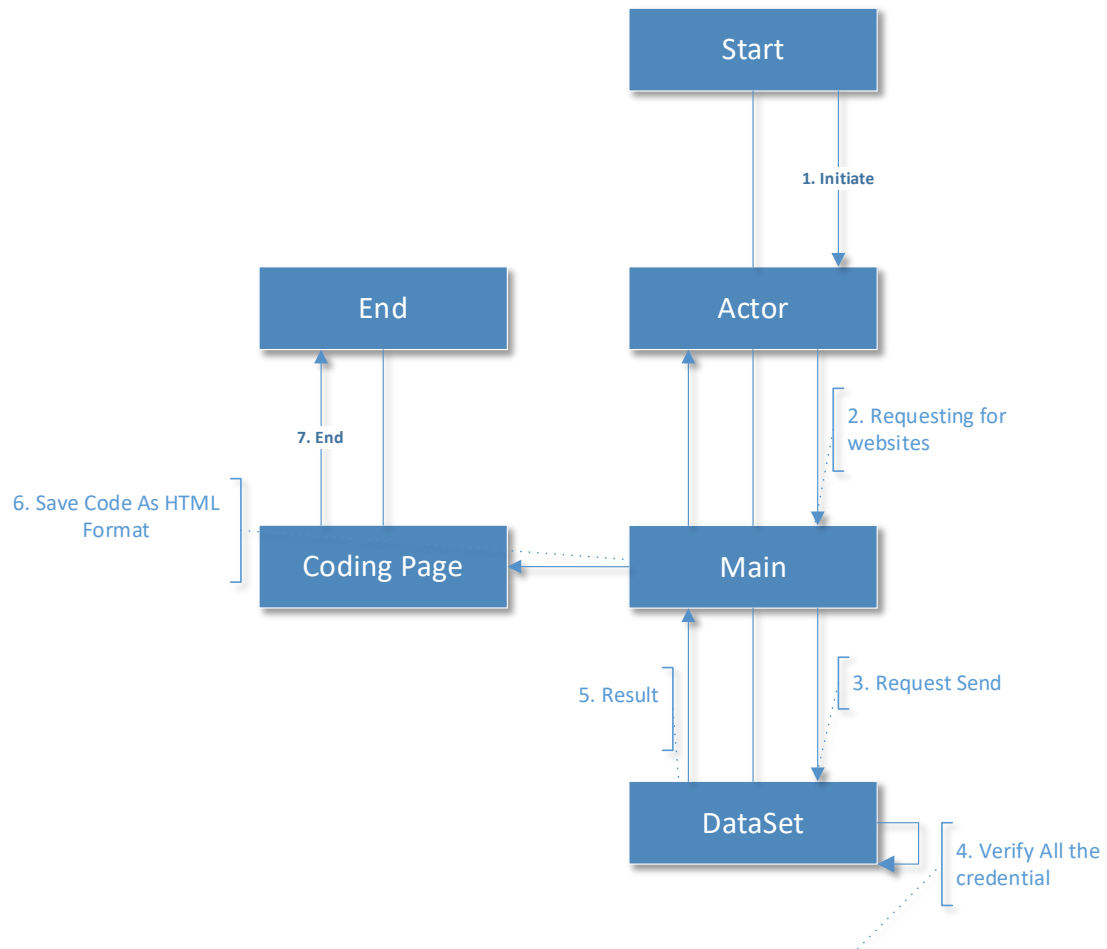
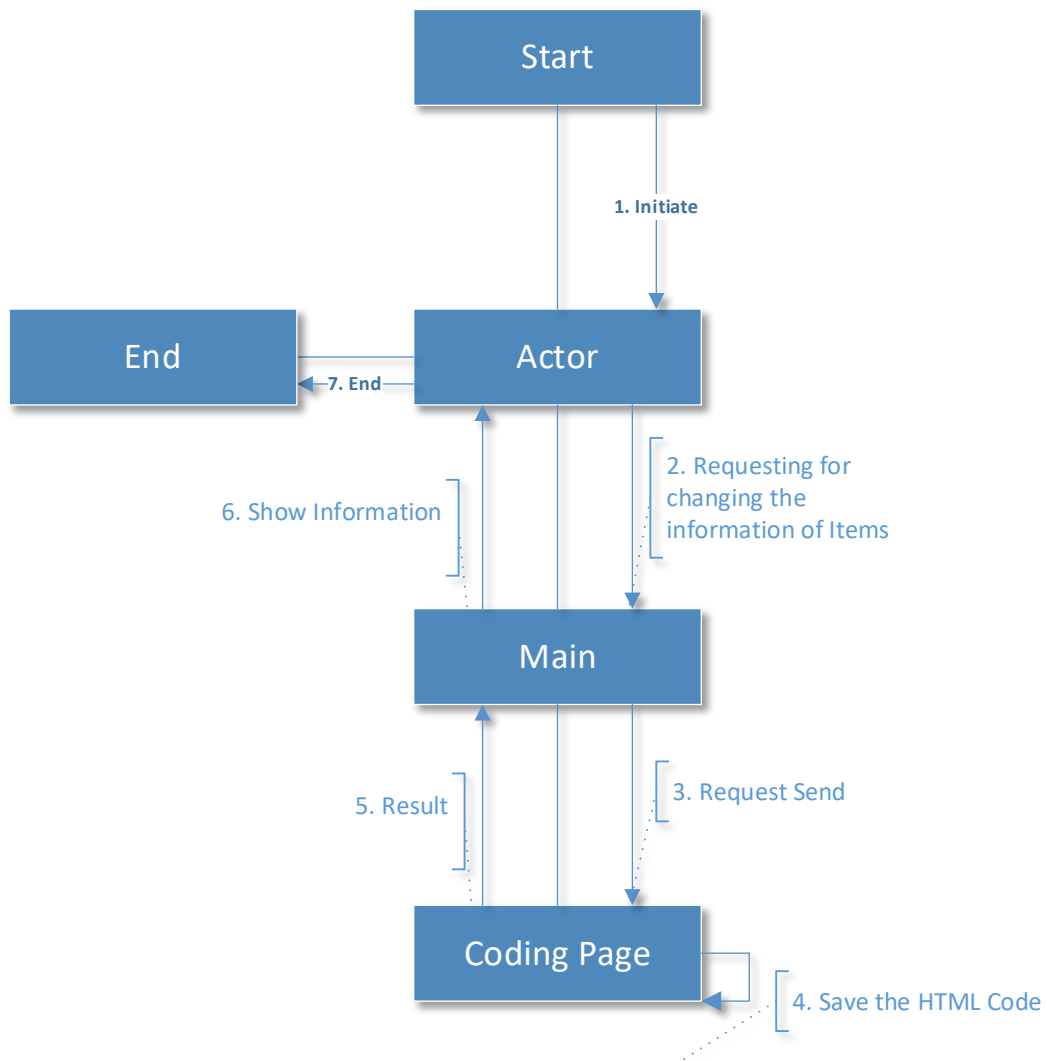
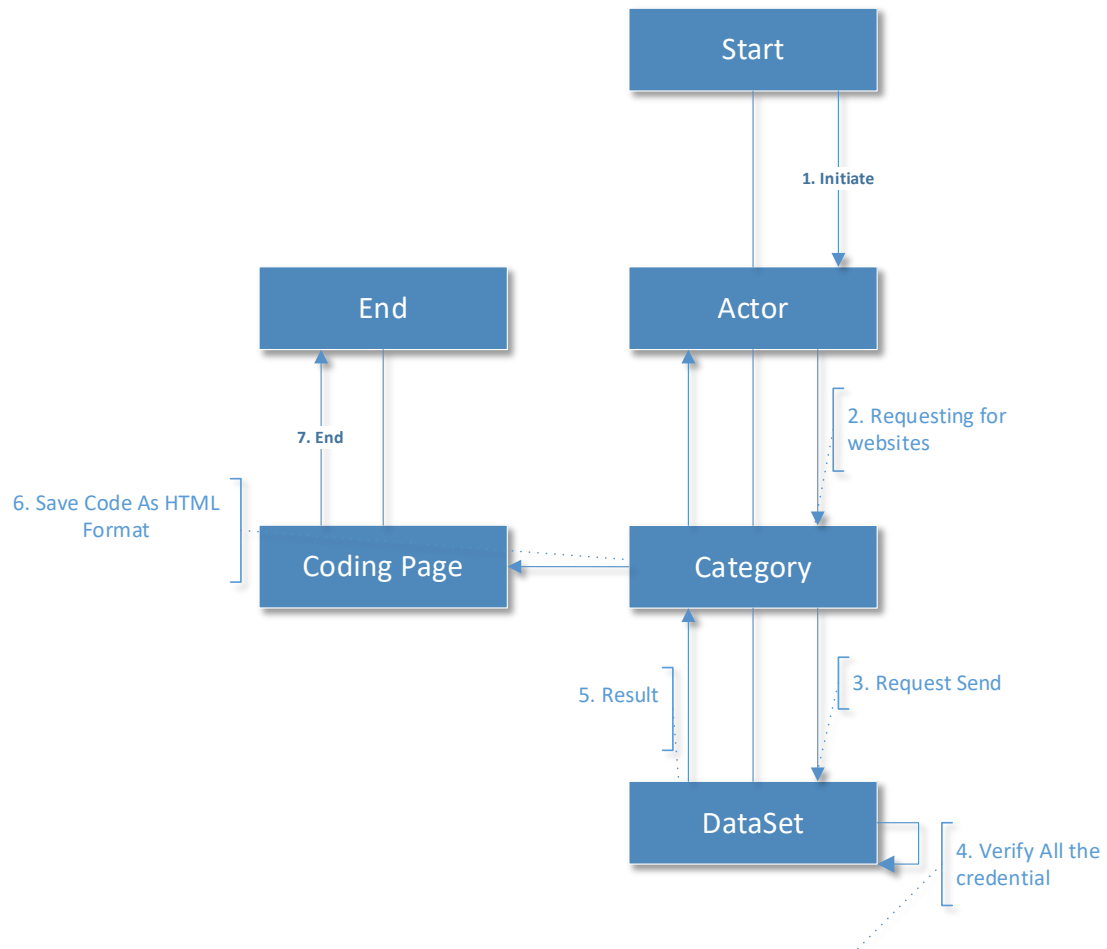


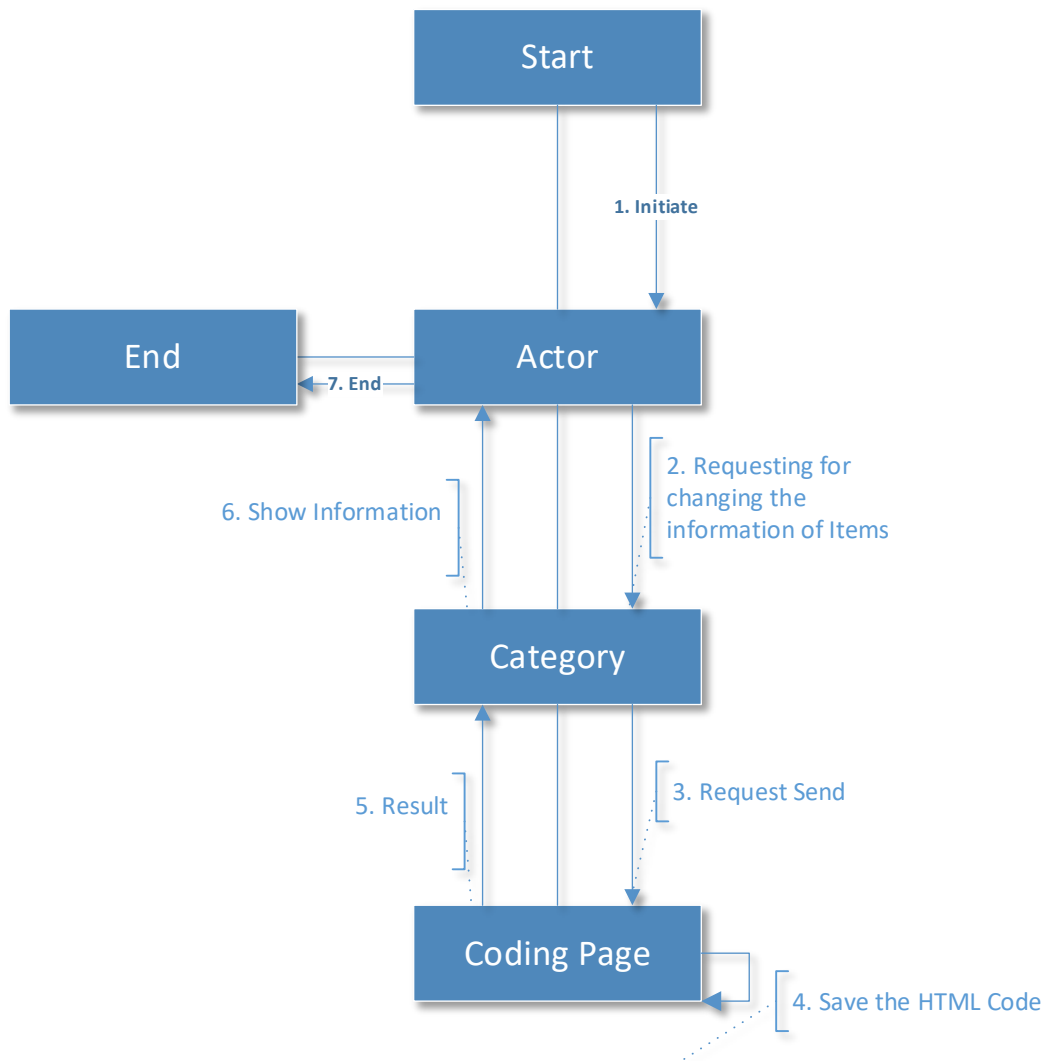
Figure 28: User End Collaboration

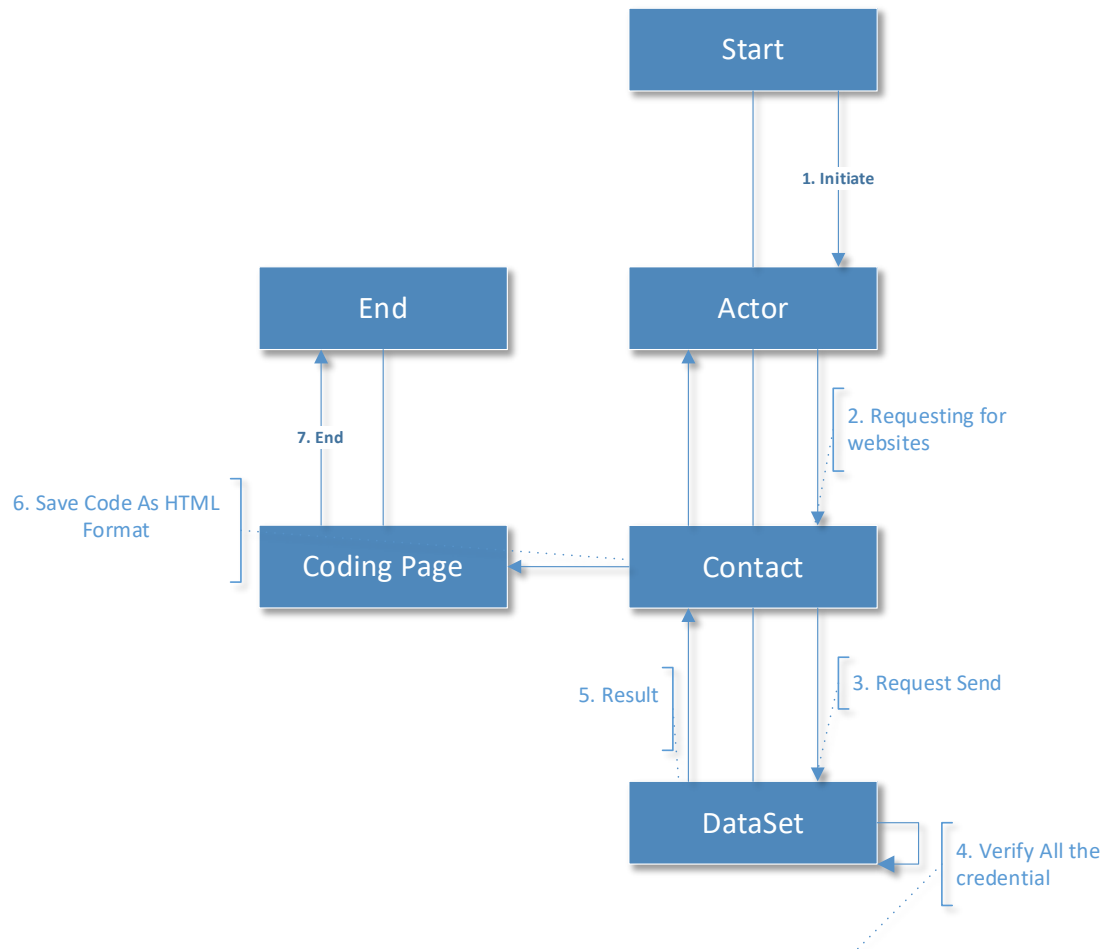


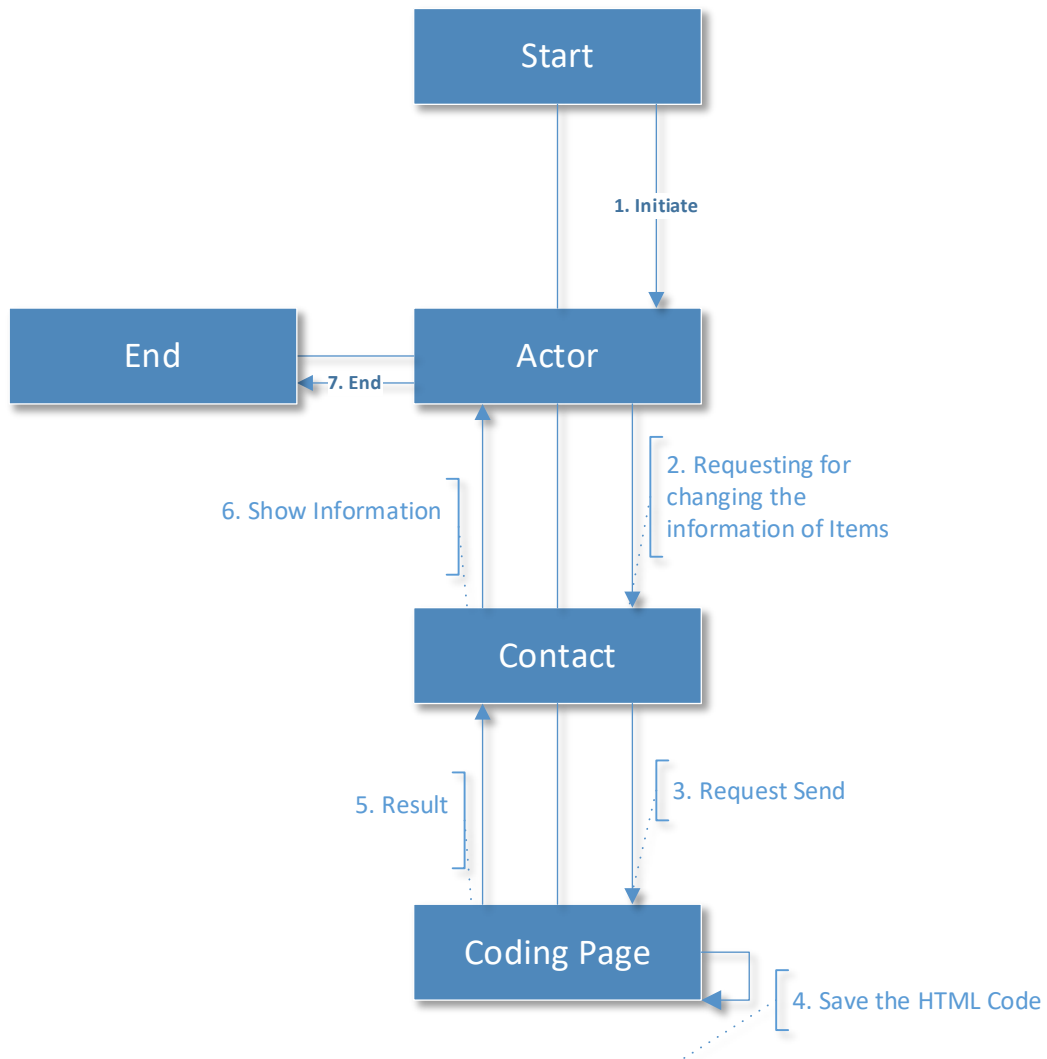


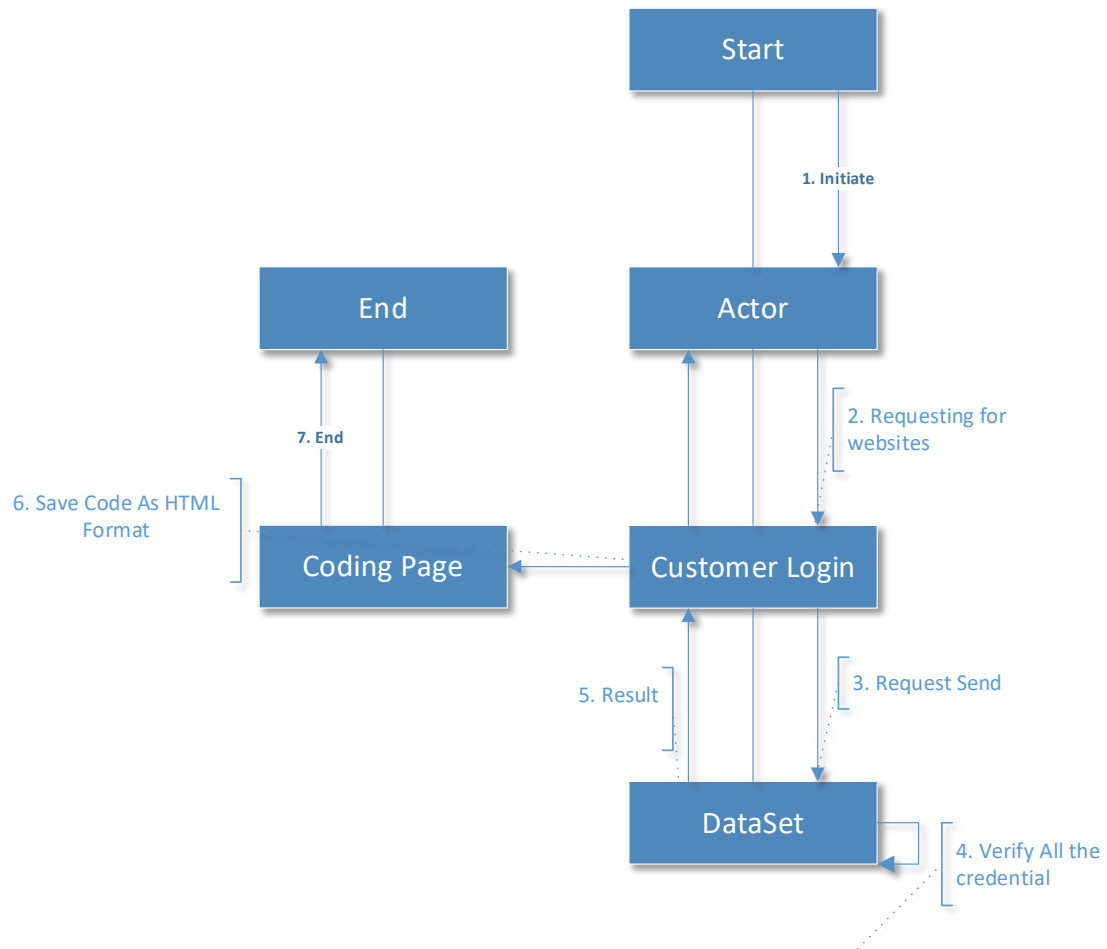


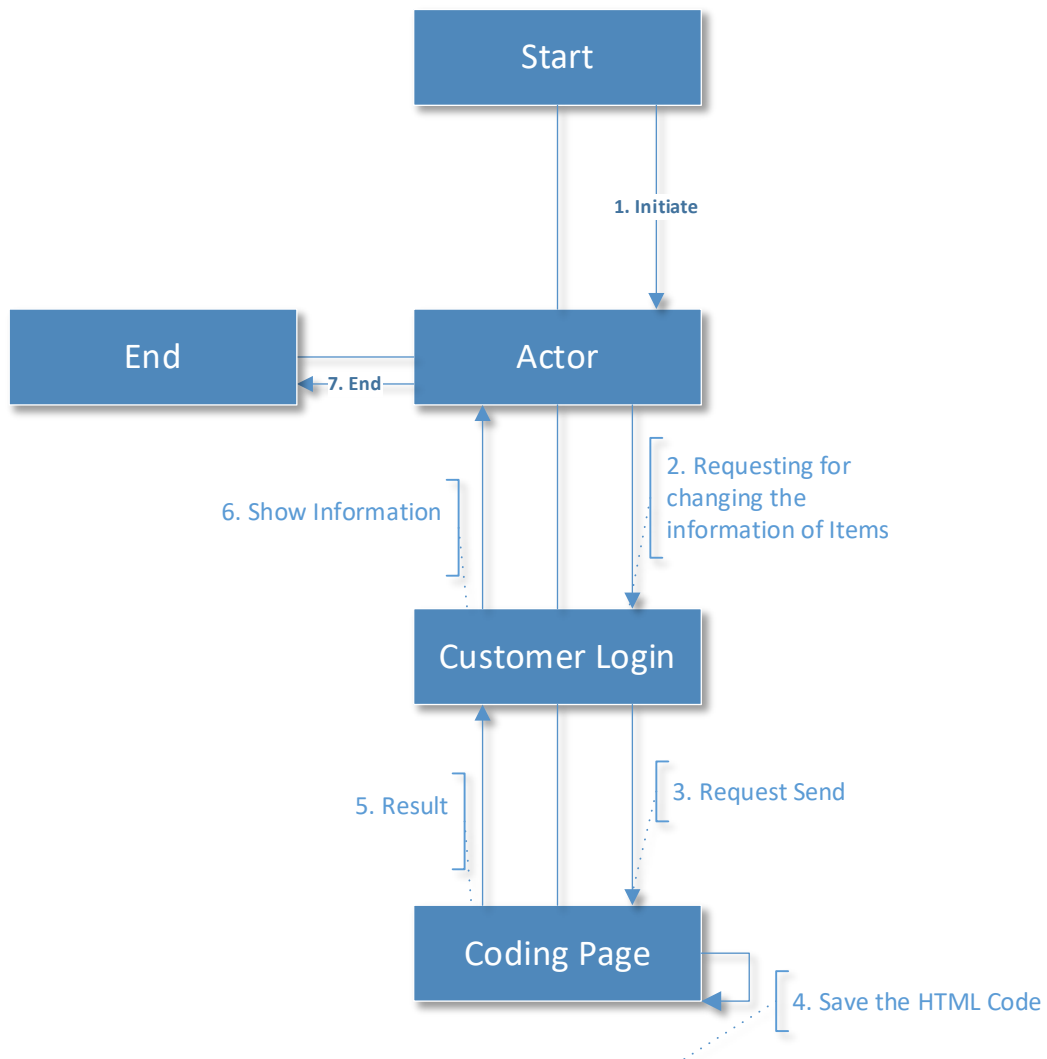


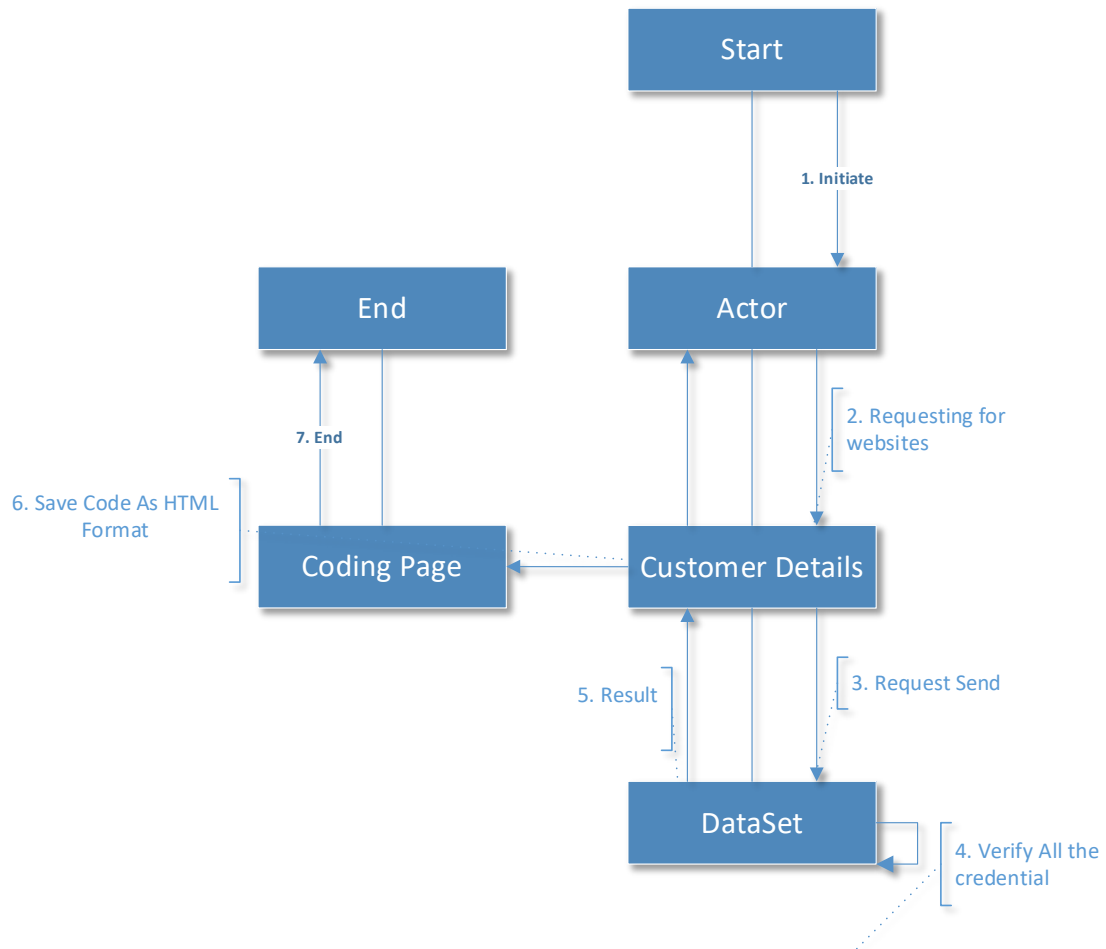


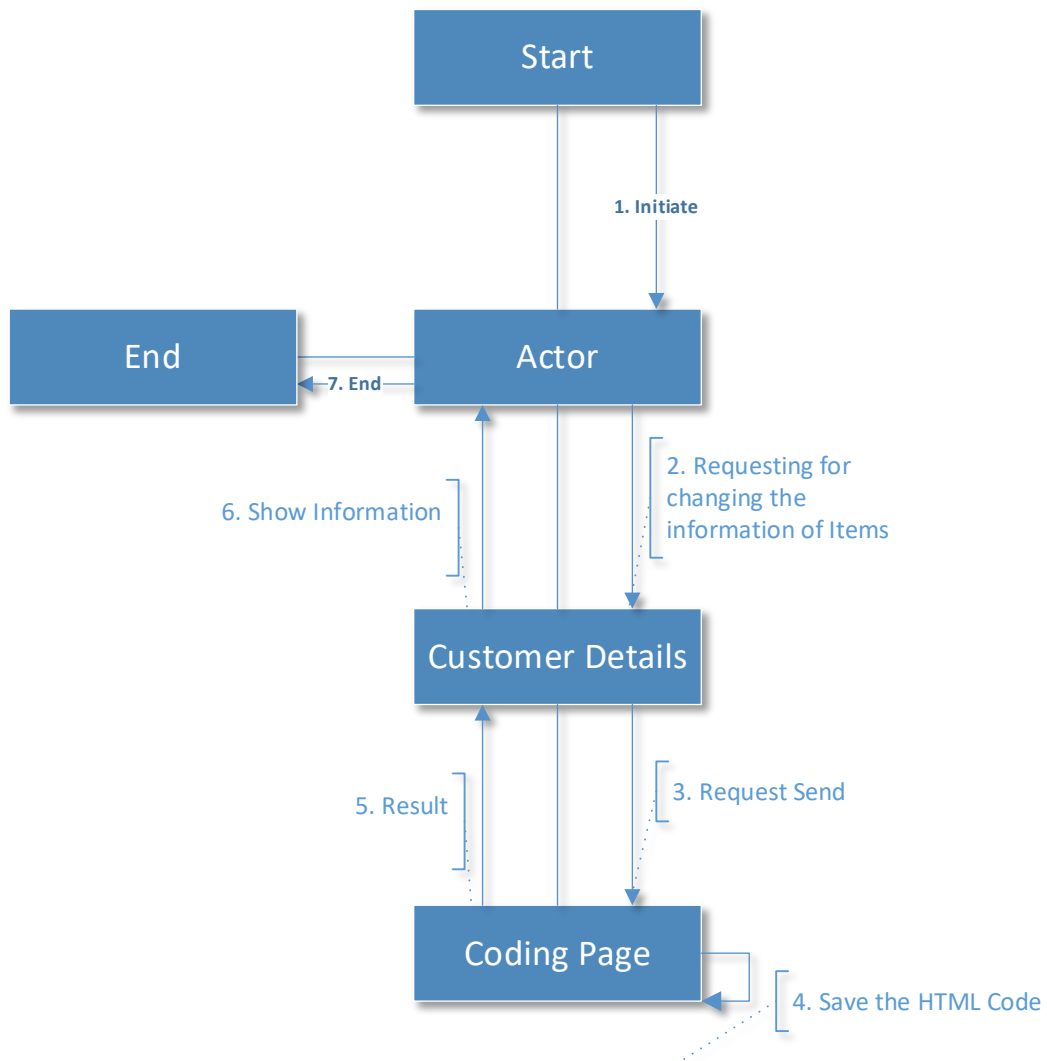


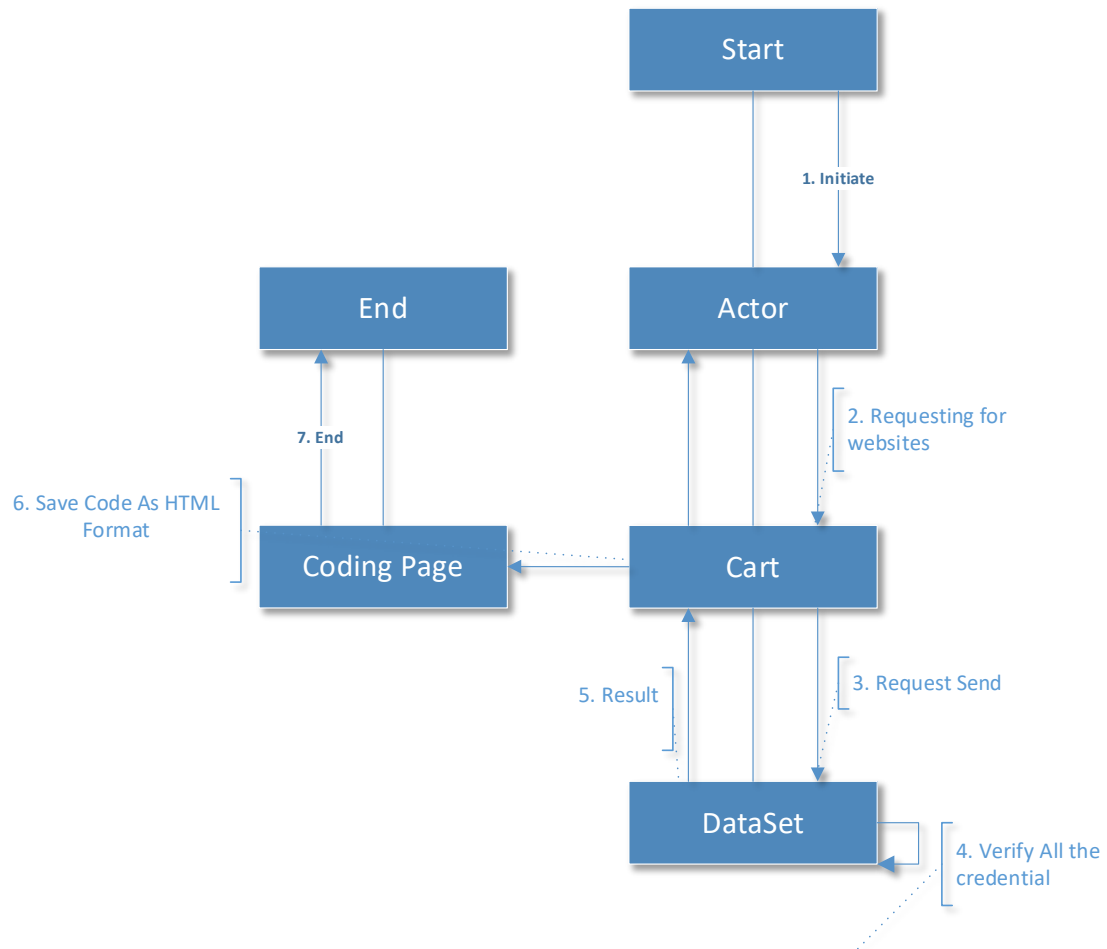


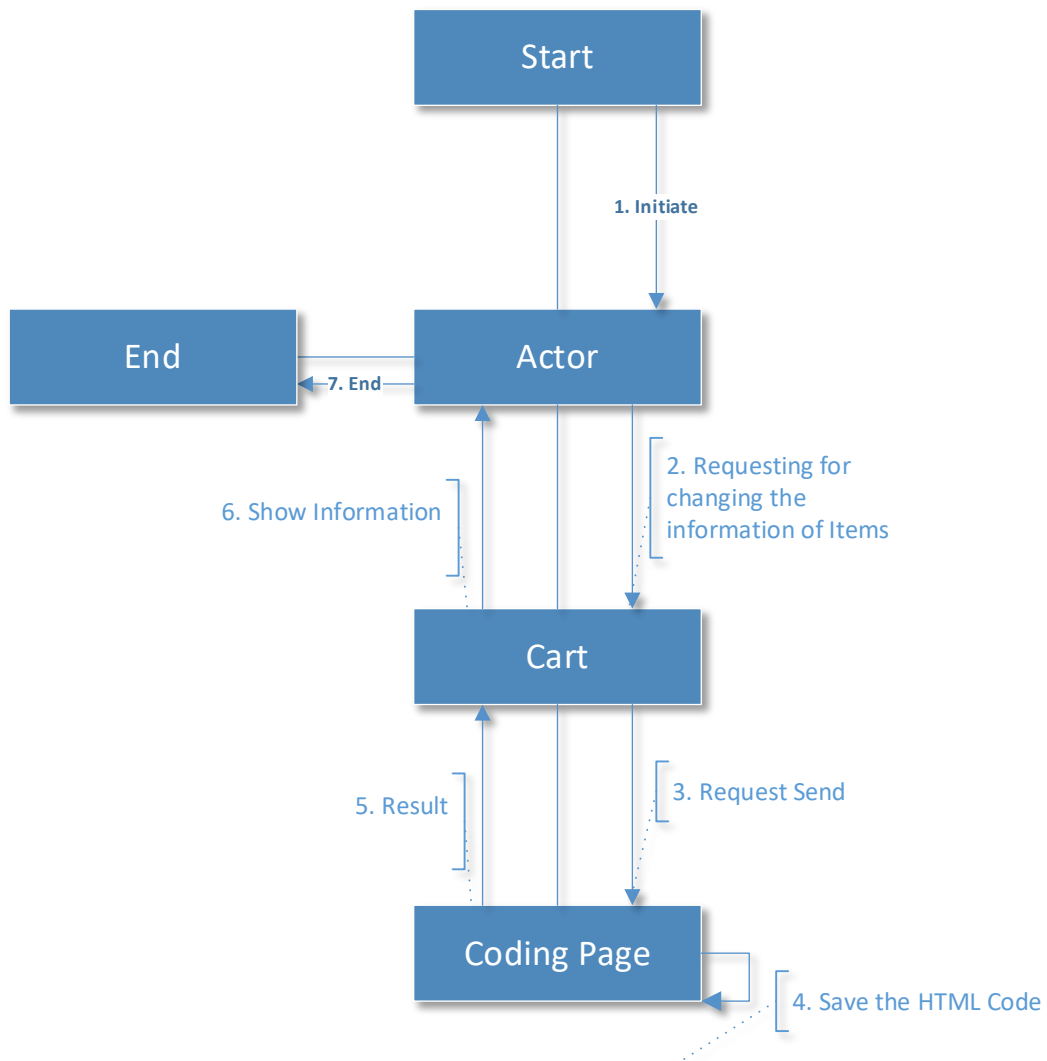


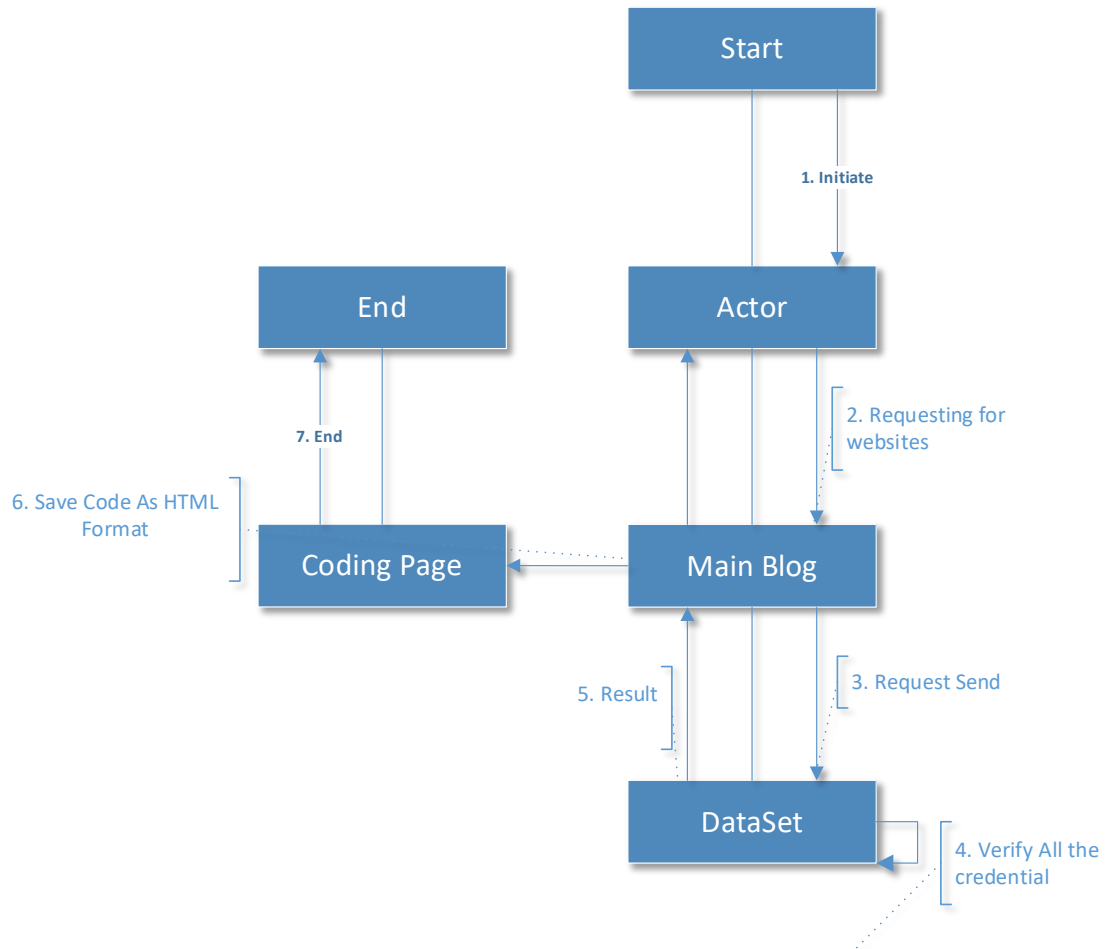


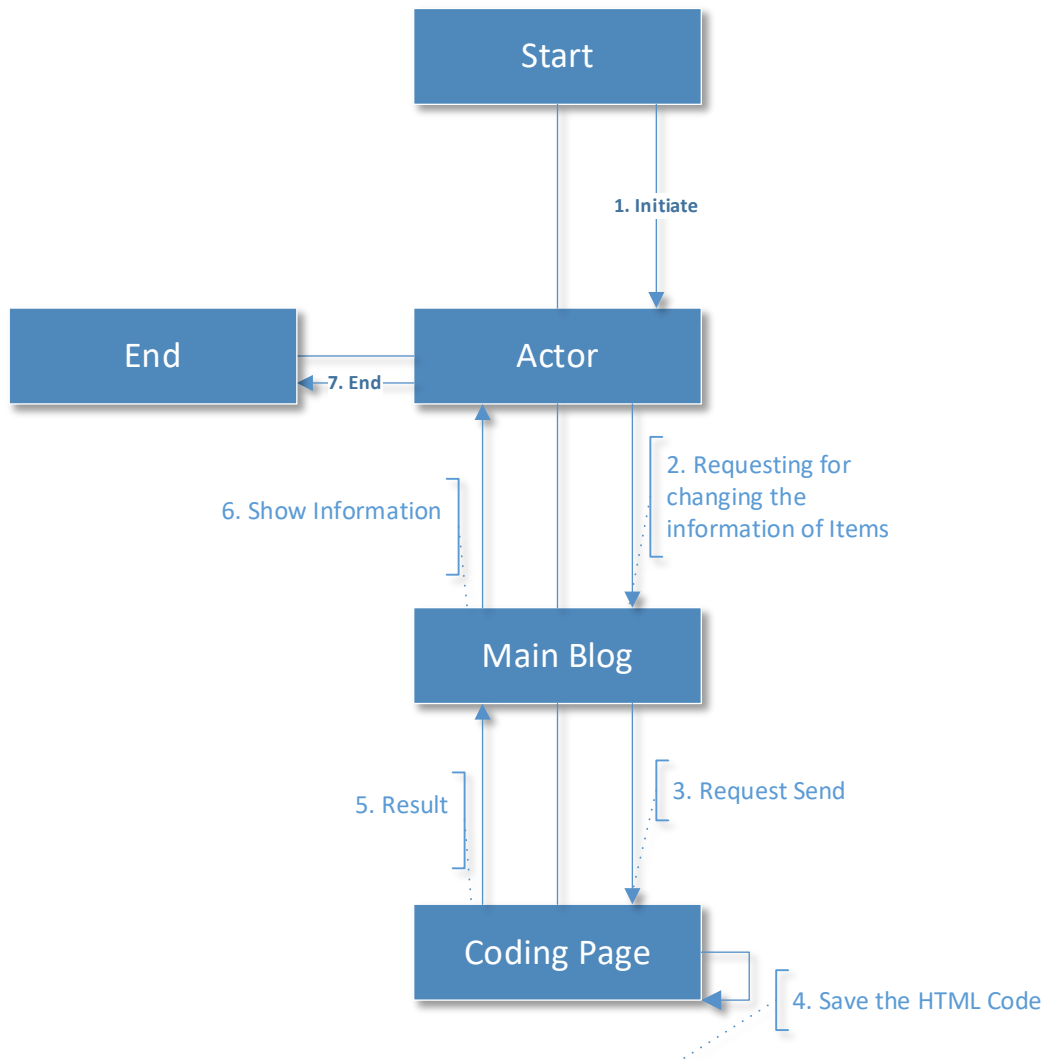


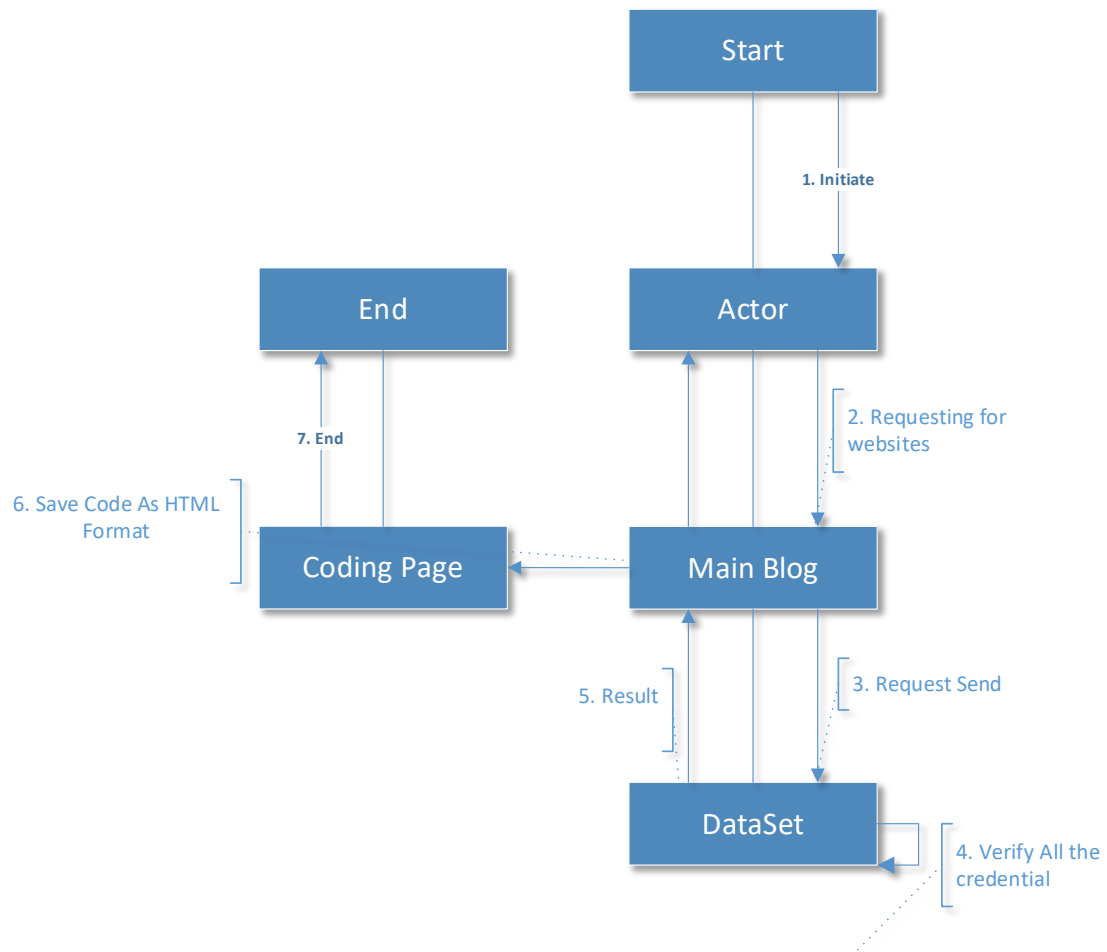


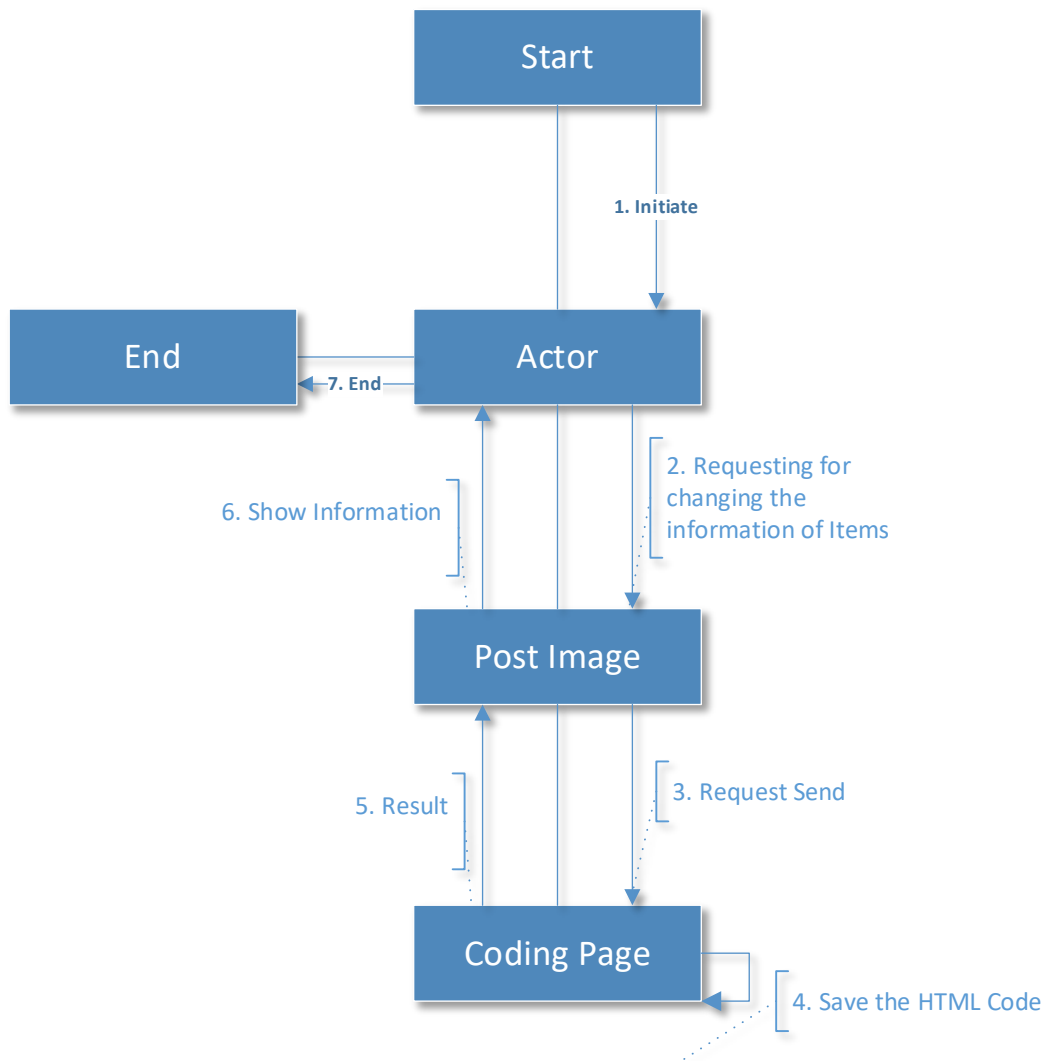


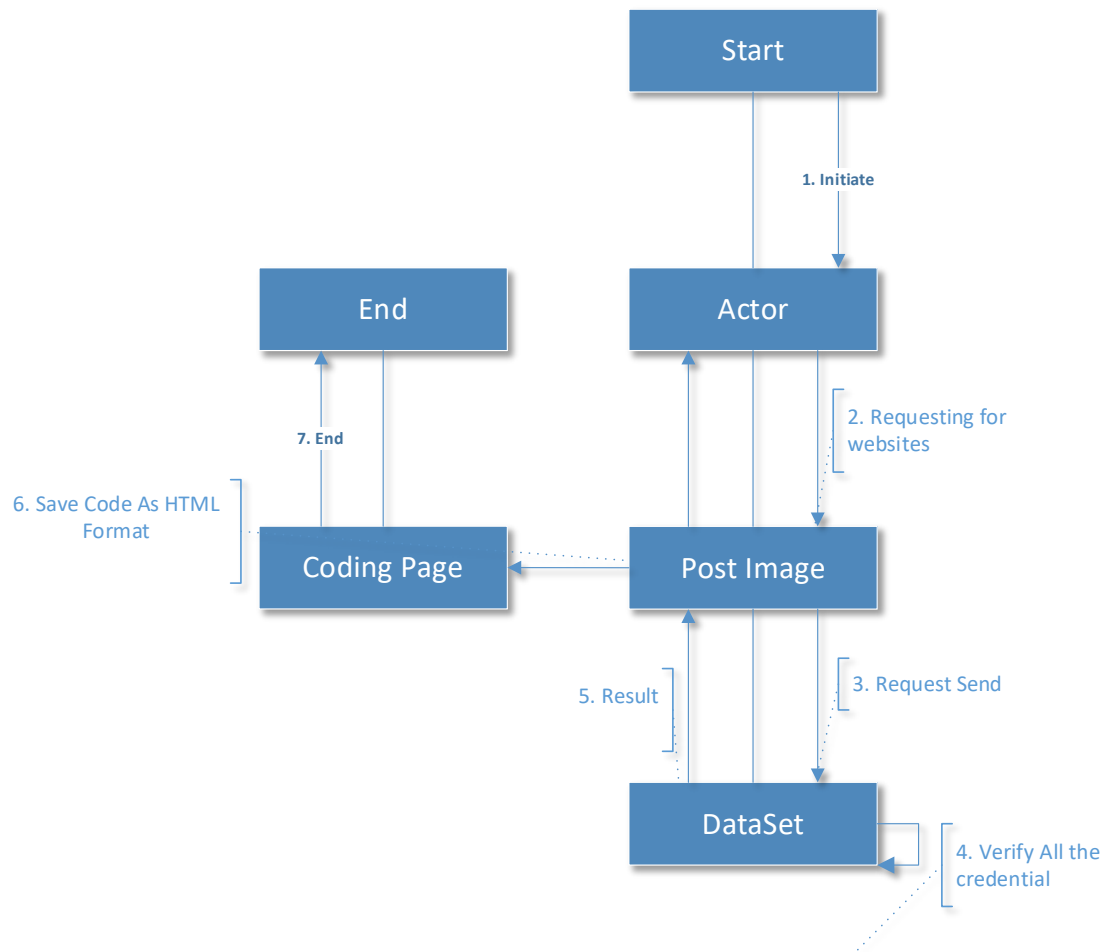


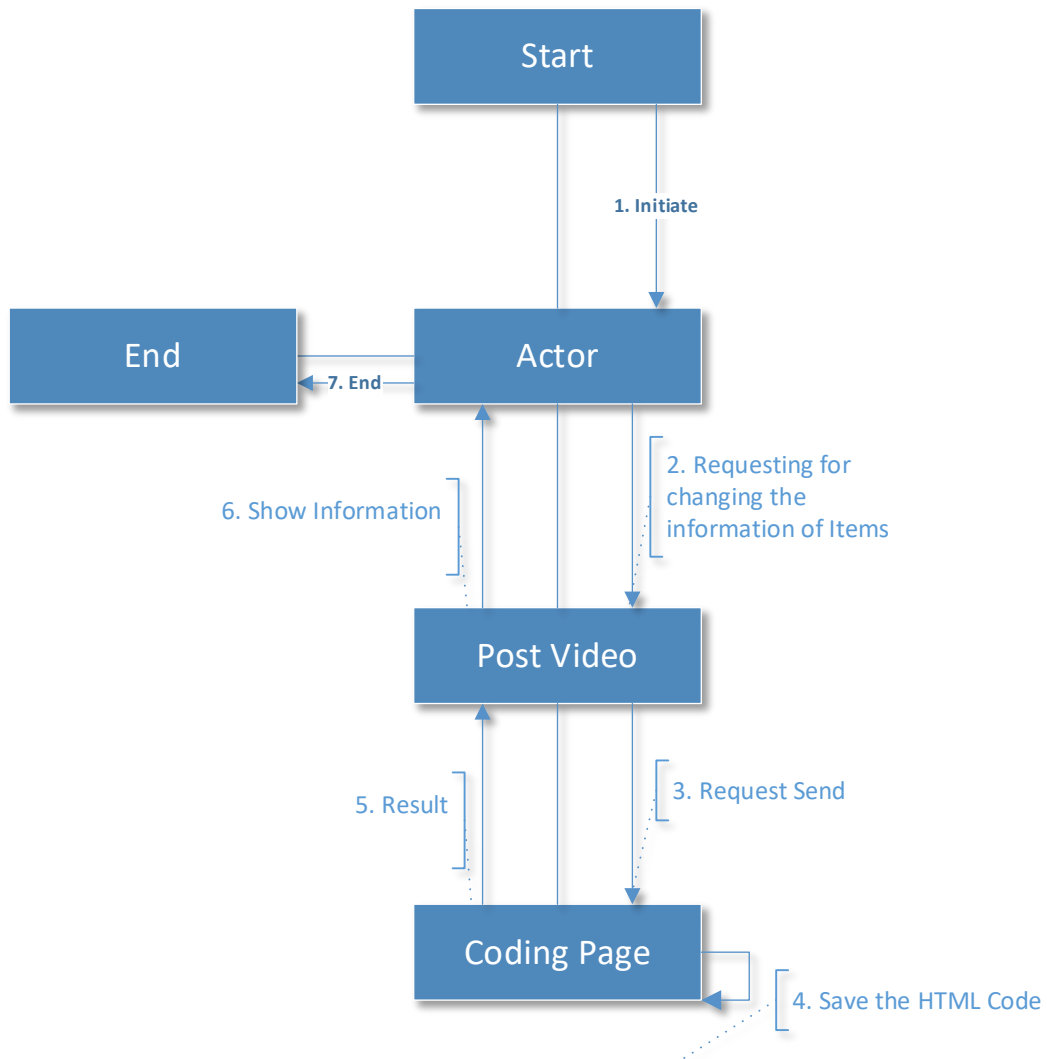


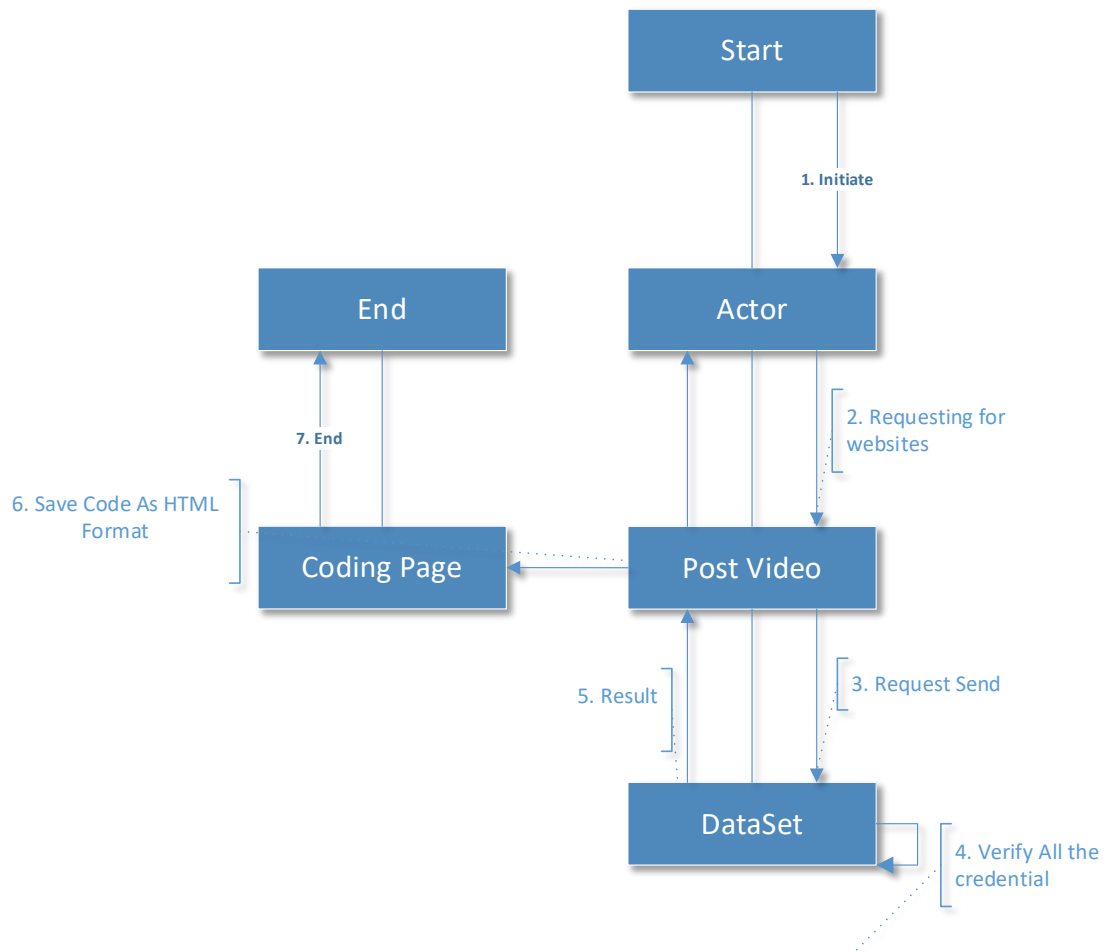


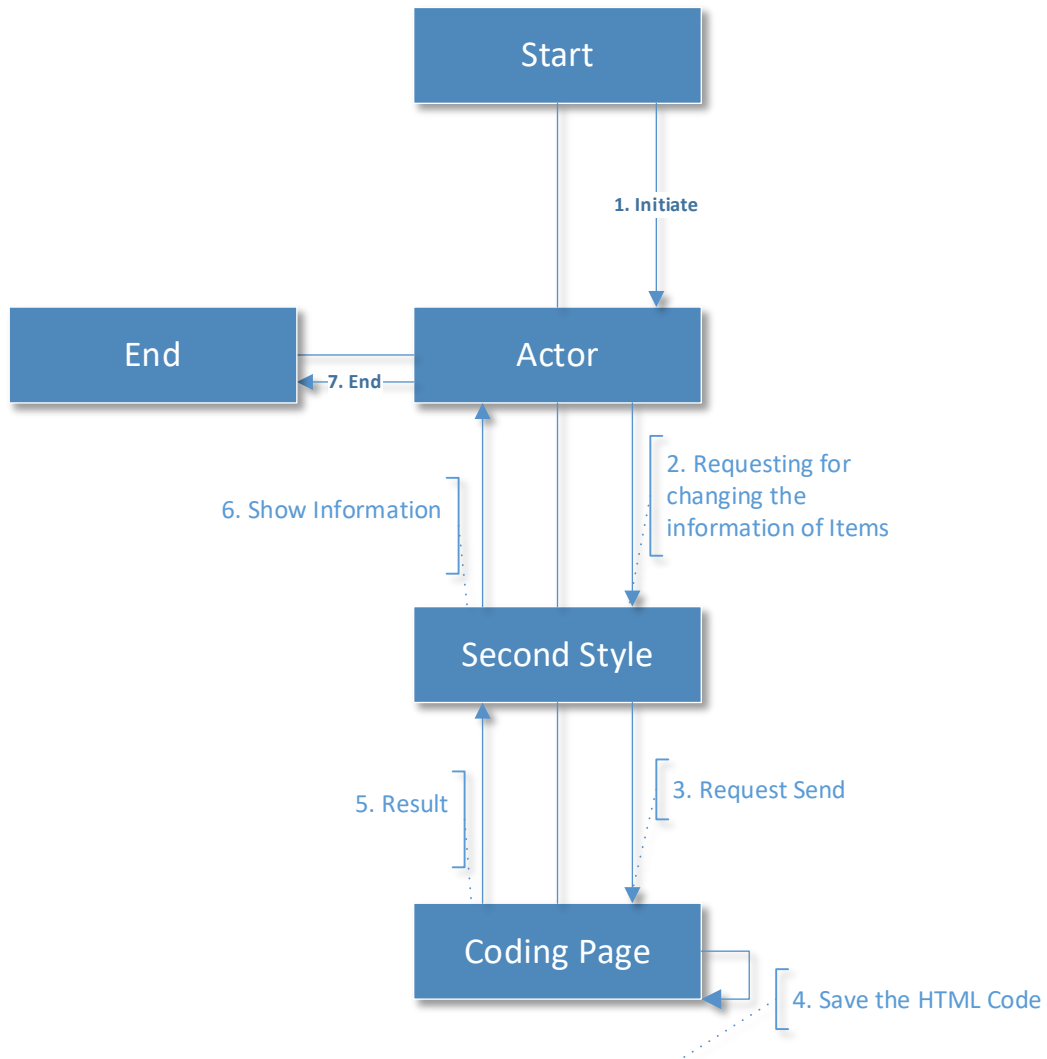


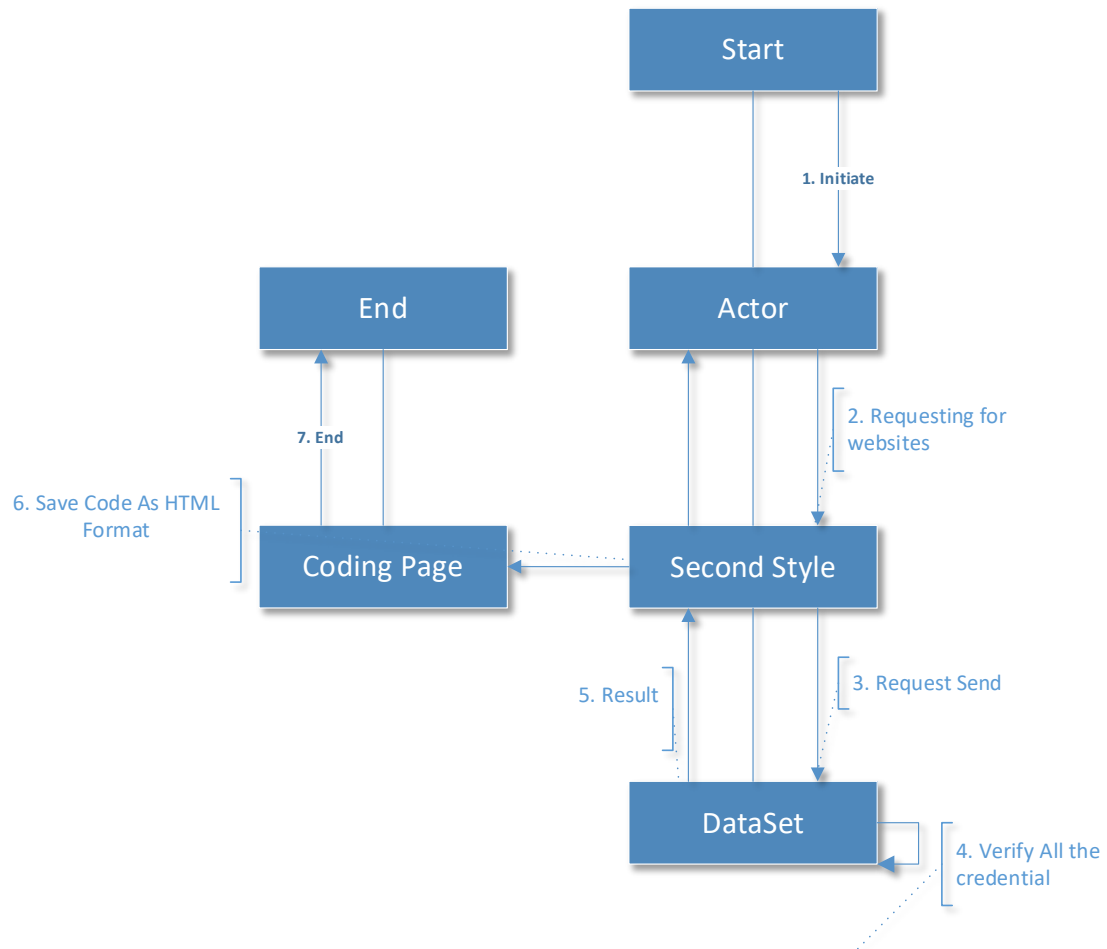


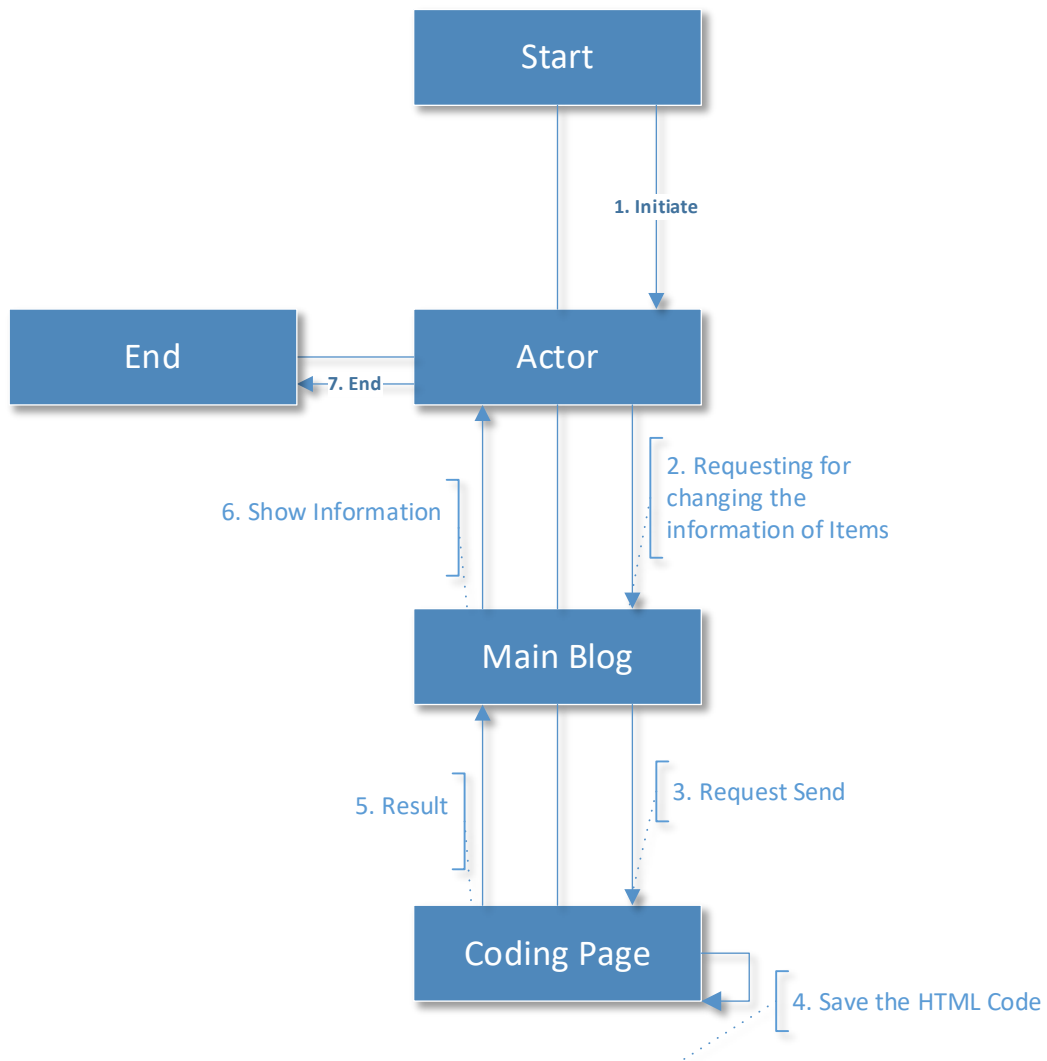


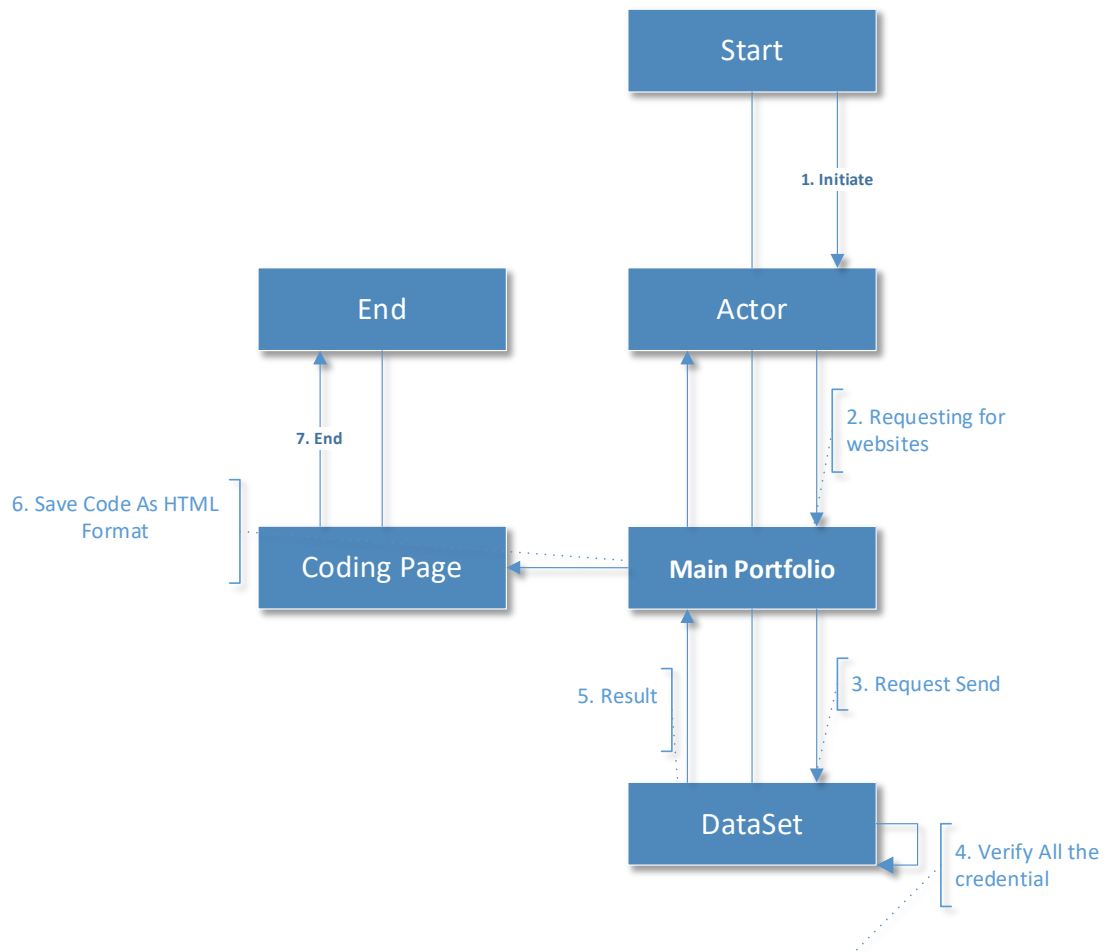


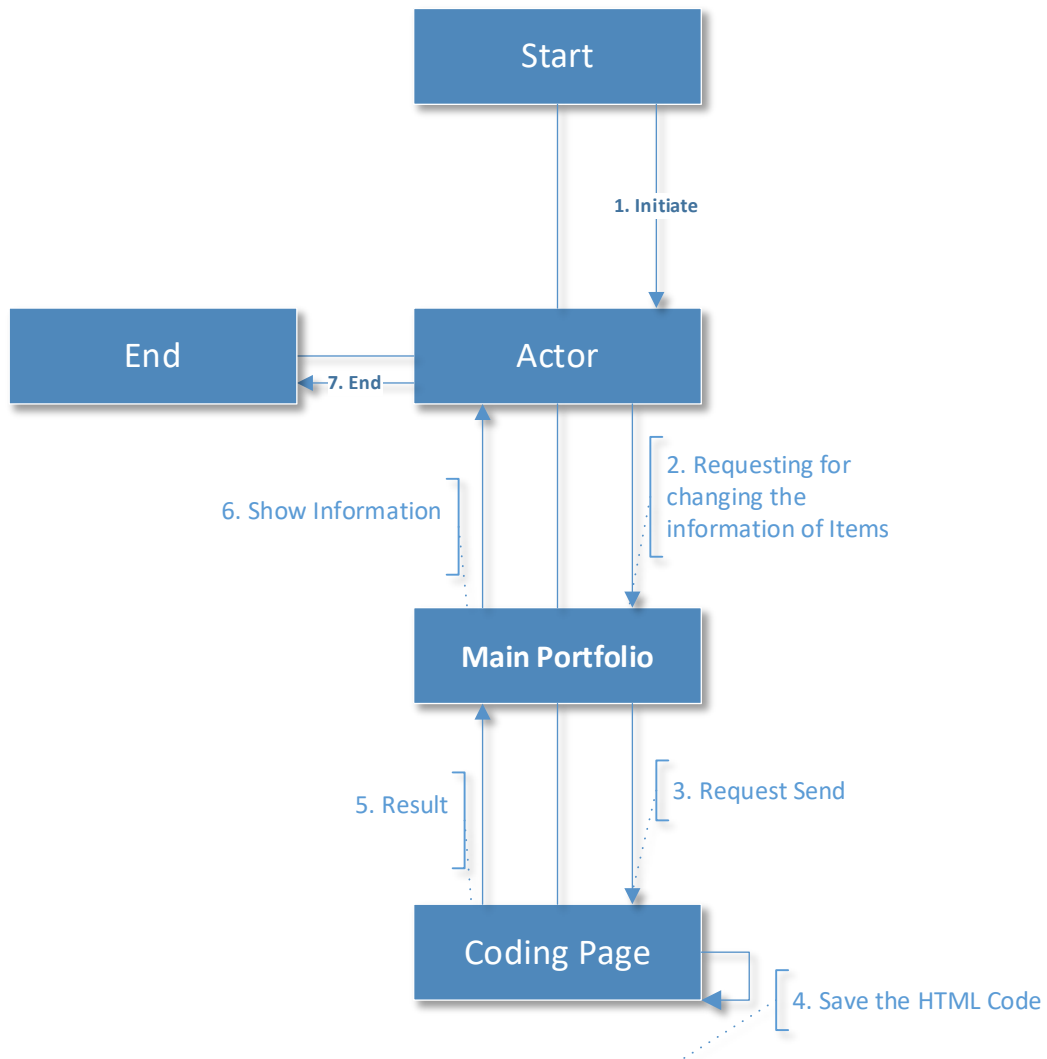


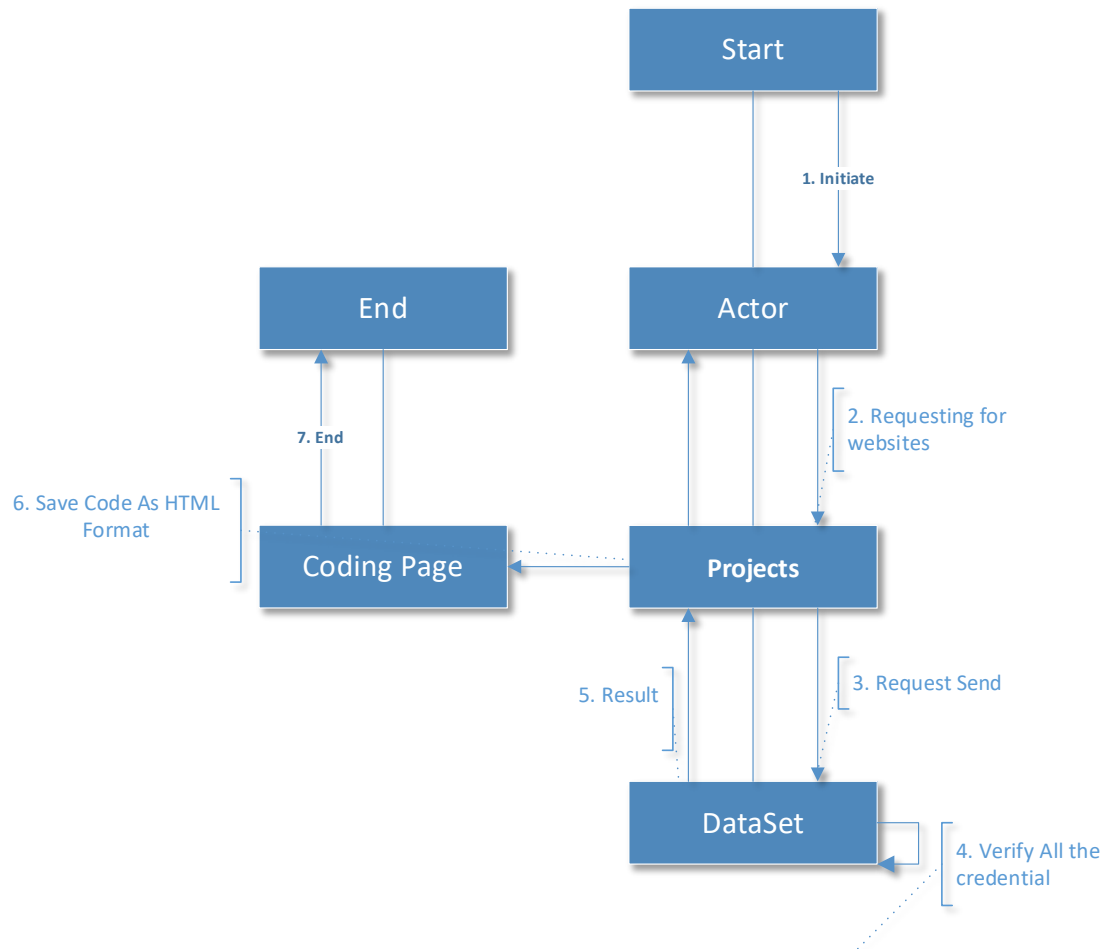


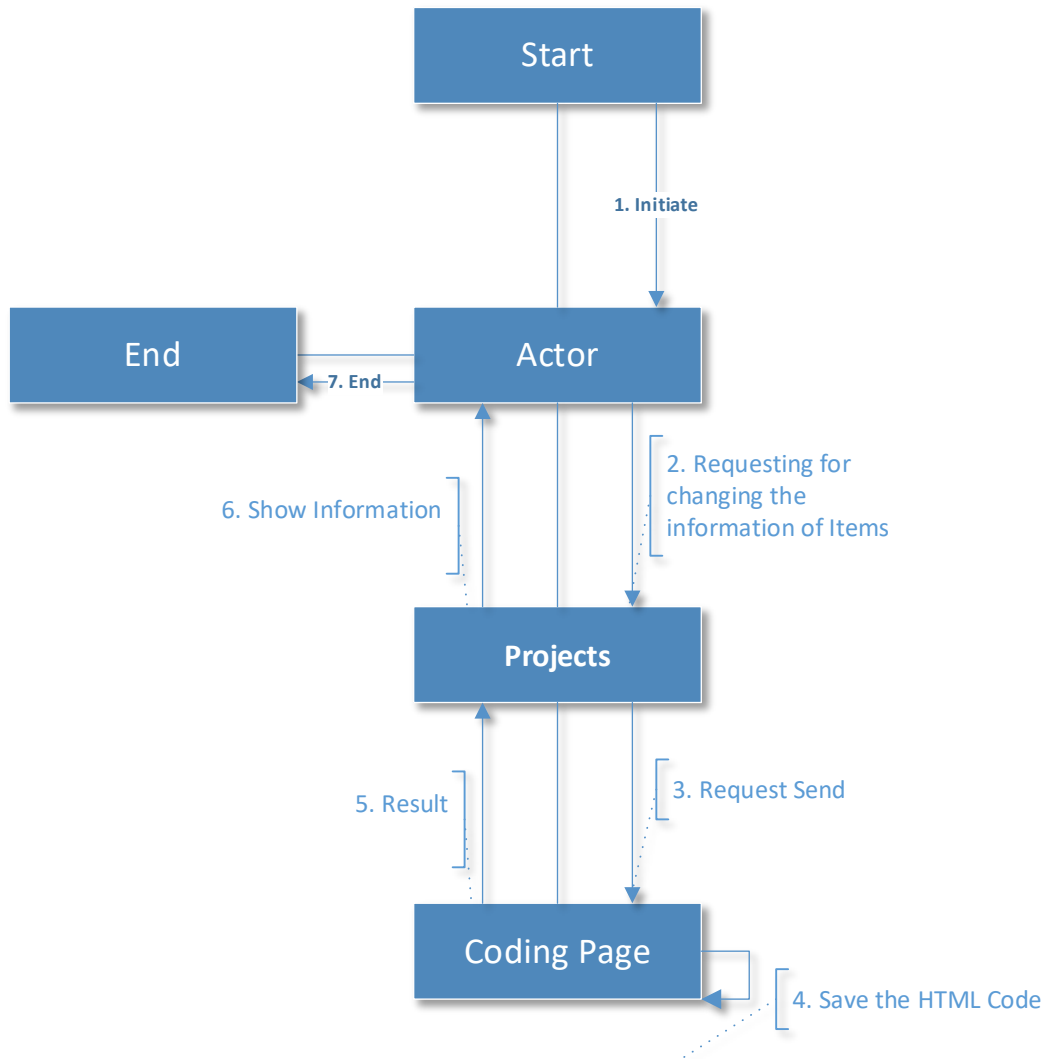


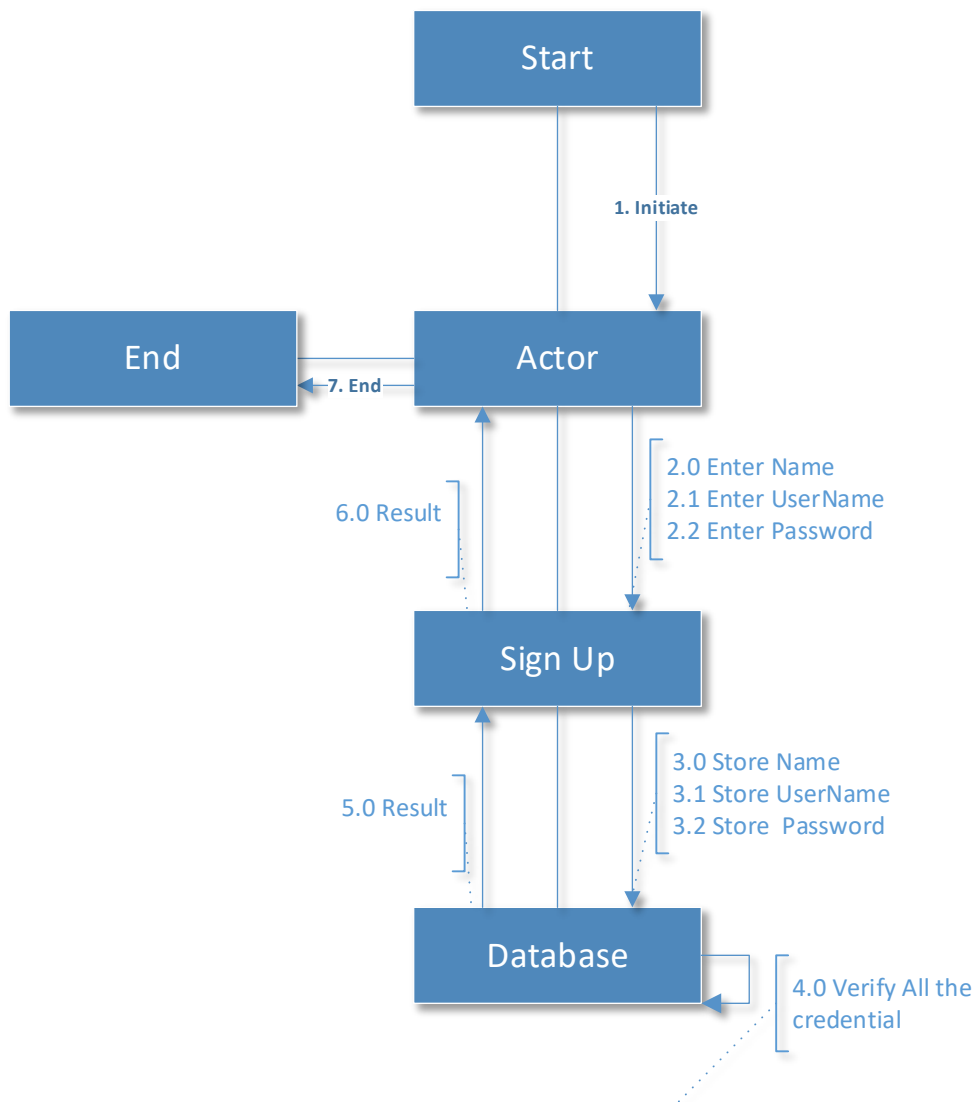


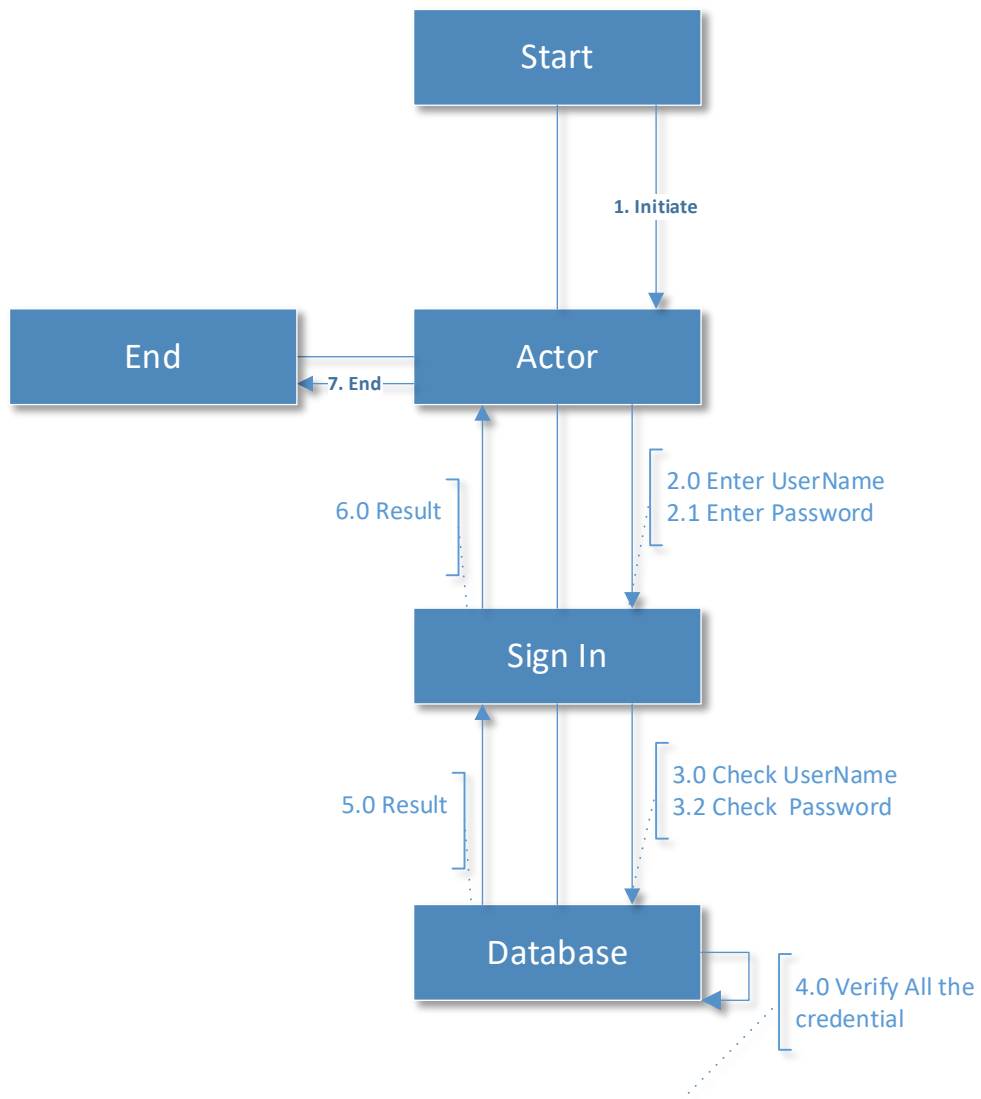


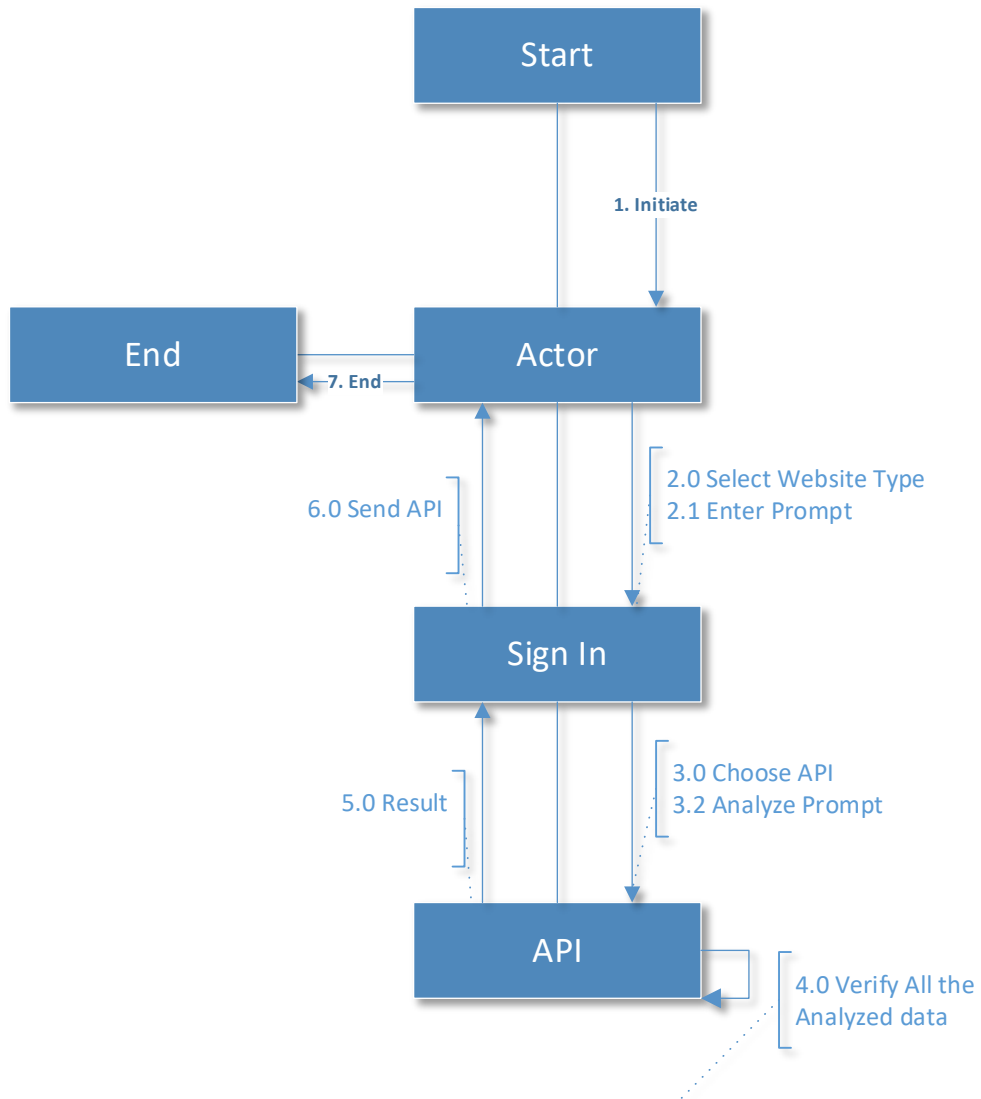


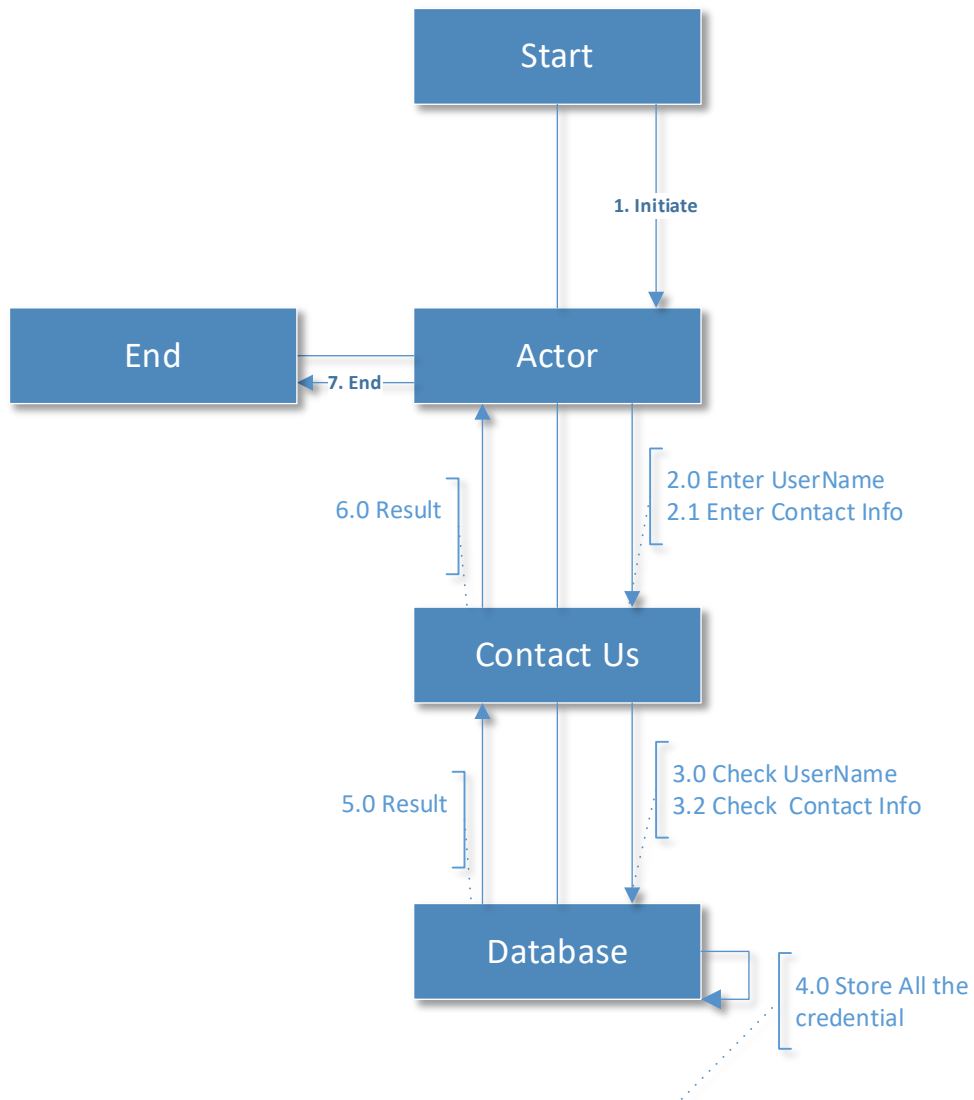












2.2 System Structure Design

2.2.1 Class Diagram

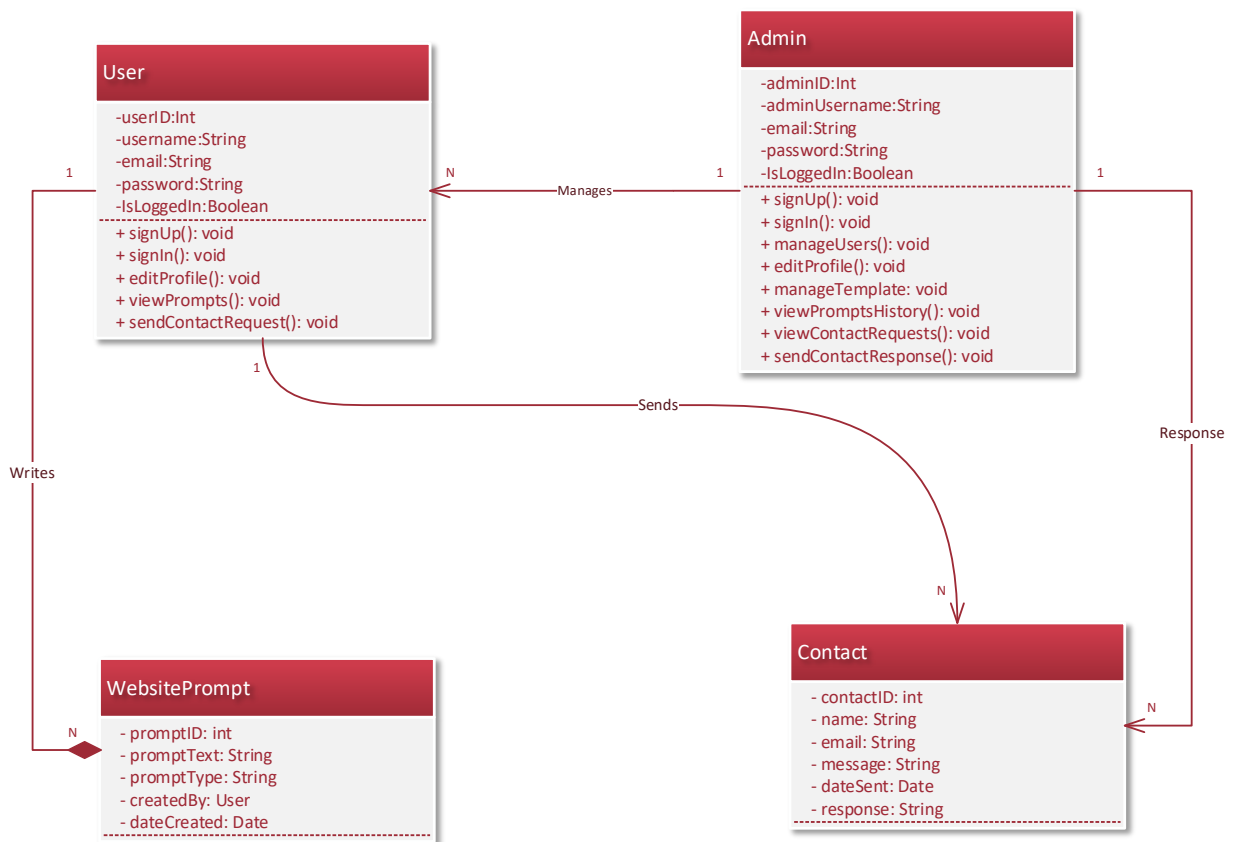


Figure 29: Class Diagram

2.2.2 Component Diagram

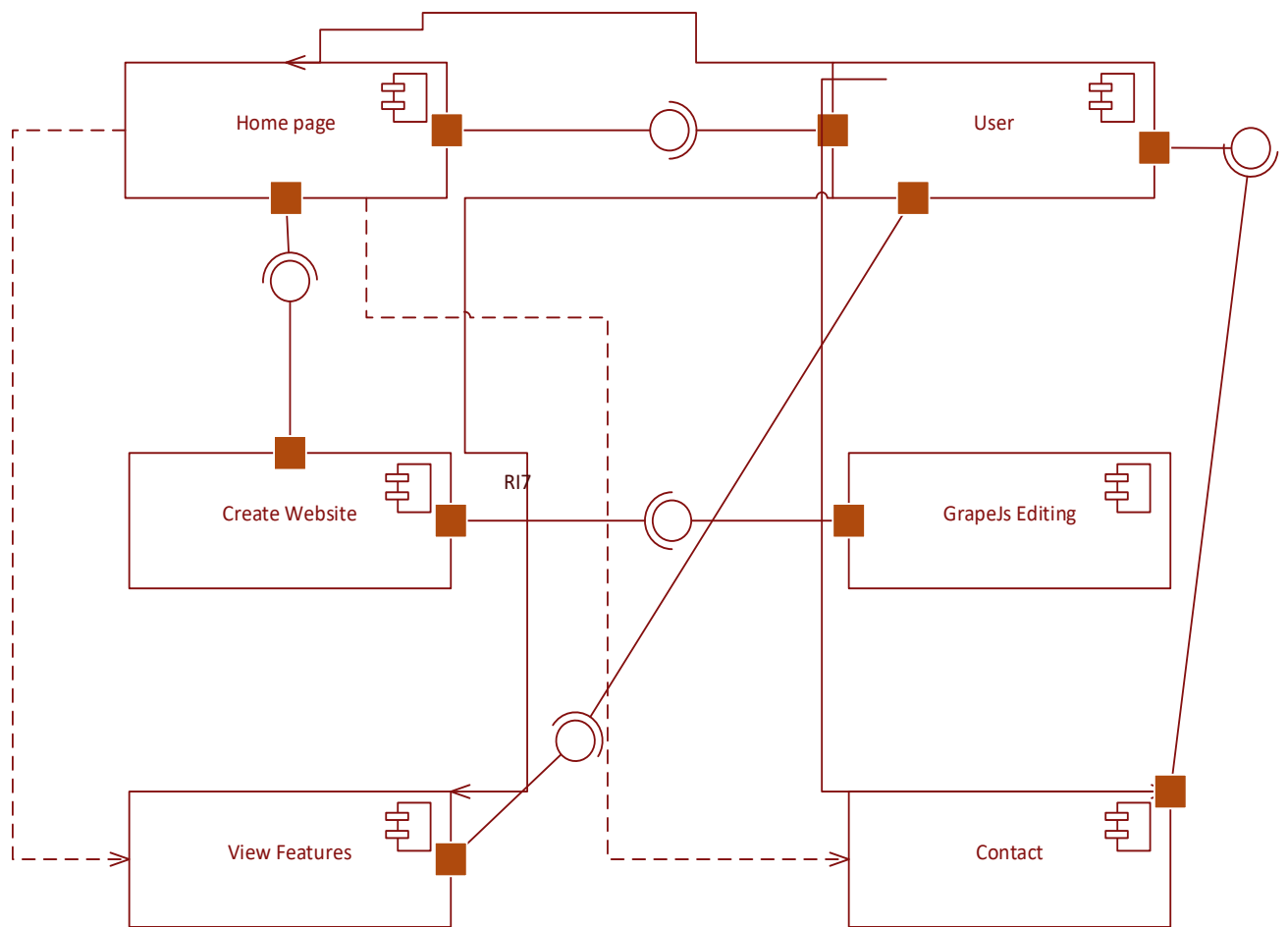


Figure 30: Component Diagram

2.2.3 Deployment Diagram

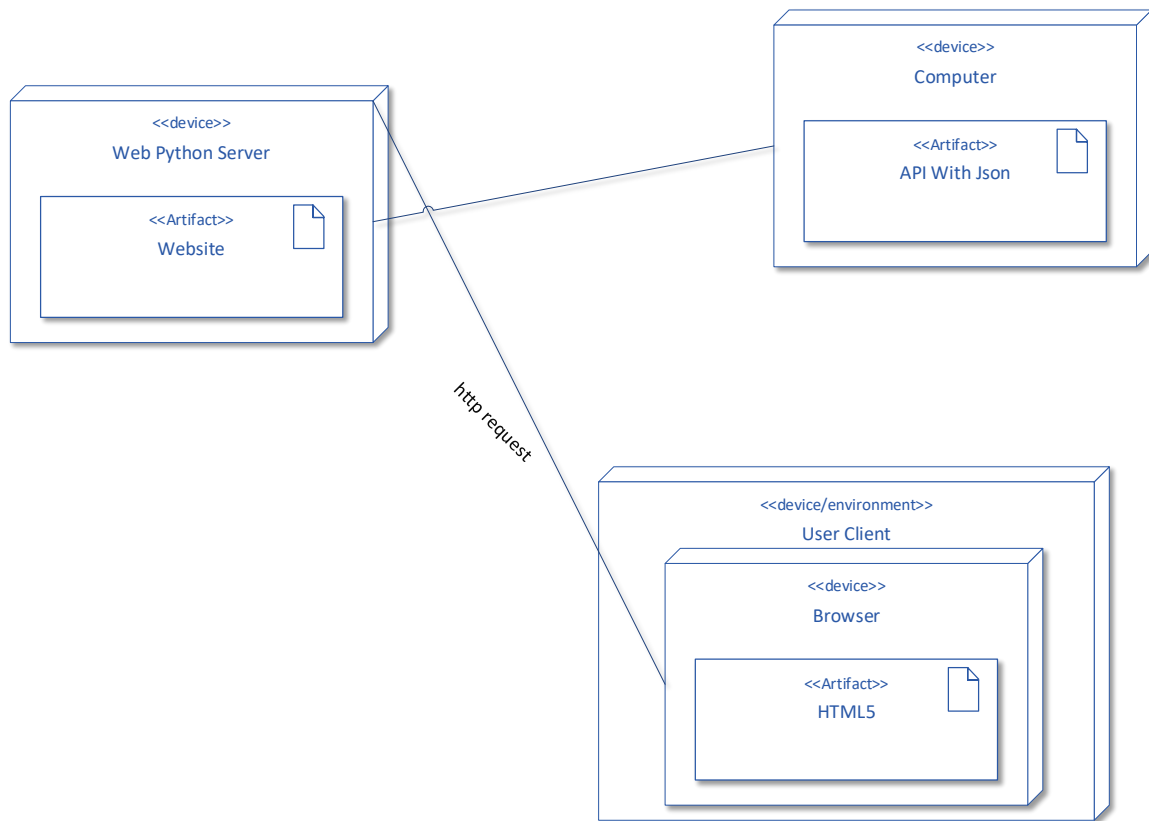


Figure 31: Deployment Diagram

2.3 User Interface Design

2.3.1 Wireframes

2.4 Database Design

2.4.1 ER Diagram

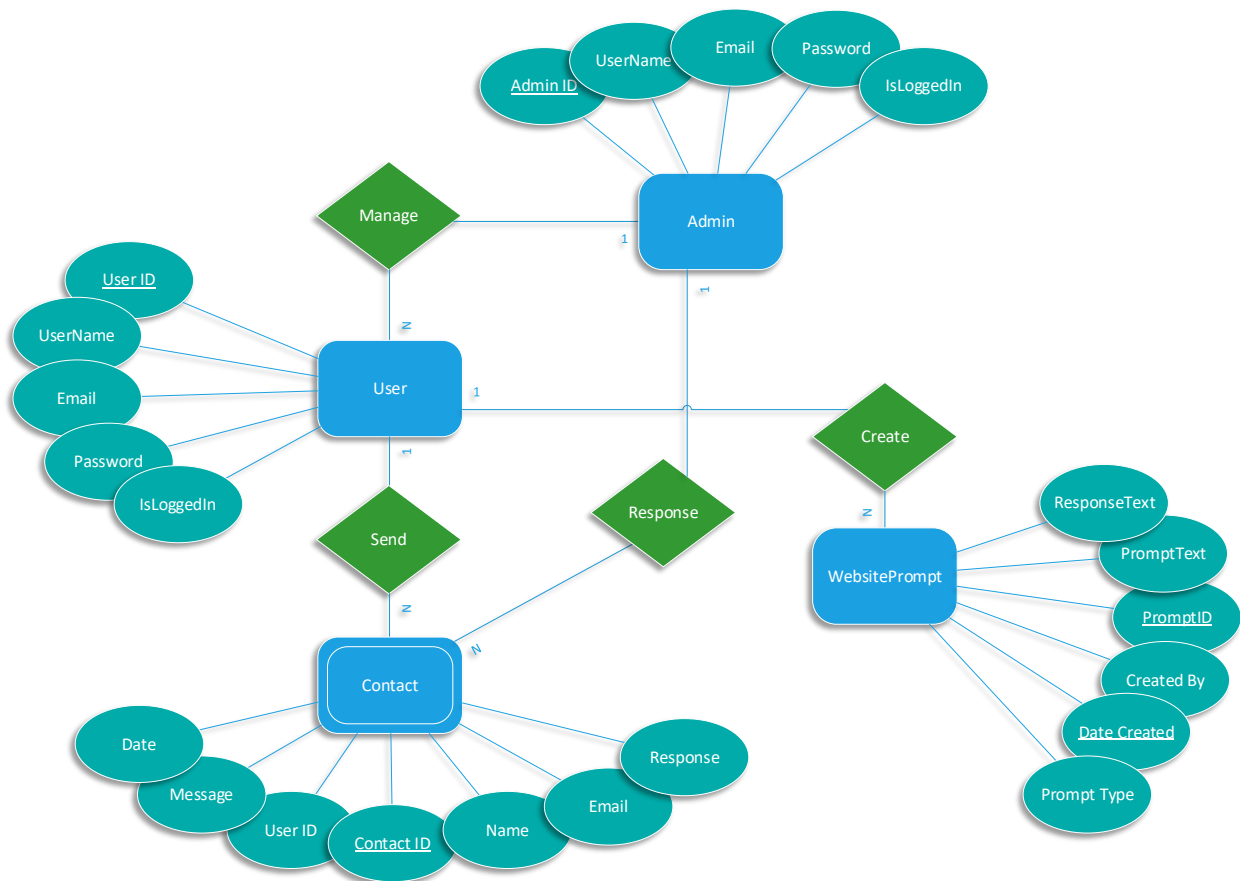


Figure 32: ER Diagram

