**SOFTWARE DESIGN SPECIFICATION**

**FERTILIZER PLANT**

**MANOJ KUMAR (U101116FCS065)**

**NIKHIL GAHLOT (U101116FCS077)**

**INDEX**

Contents

**1. Introduction**

**1.1 Purpose of this document**

**1.2 Scope of the development project**

**1.3 Definitions, acronyms, and abbreviations**

**1.4 References**

**1.5 Overview of document**

**2. Conceptual Architecture/Architecture Diagram**

**2.1 User interface issues**

**3. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)**

**3.1 Logical Architecture Description**

**3.1.1 Class Diagram Explanation**

**3.1.2 State Diagram Explanation**

**3.1.3 Sequence Diagram Explanation**

**4.0 Execution Architecture**

**4.1 Reuse and relationships to other products**

**5.0 Design decisions and tradeoffs**

**6.0 Pseudo code for components**

**7.0 Appendices (if any)**

**The Software Design Specification**

1. **Introduction**

The Software Design Specification is a document that will provide the overall guidance of the architecture of the software project to software development team. It provides the information how software should be built. It include the graphical representation of the function or methods using the use case, sequence diagram, state diagram, class diagram or other supportive requirement information.

* 1. **Purpose of this Document**

The purpose of this SDS document is to provide a architecture/blueprint of the software. It contains the information of all the classes, function we are providing and methods we are implementing. This Design document will specify to the developers what is need to be created and how it is going to be implemented.

* 1. **Scope of the Development Project**

This product provides information to the user about the fertilizers and helps them to buy the product they want. It would be very effective and easy way for the users to get fertilizers home delivered. It will also provide a contact option if someone has any problem, if user doesn’t know how to use the product they can contact to the organization helpline number to get the information or help they need.

* 1. **Definitions, acronyms, and abbreviations**

SDS – Software Design Specification

IEEE - Institute of Electrical and Electronics Engineers

* 1. **References**
* IEEE SDS Template
  1. **Overview of the Document**

According to the IEEE document the SDS is divided into seven subparts which will conclude the complete overview of the document.

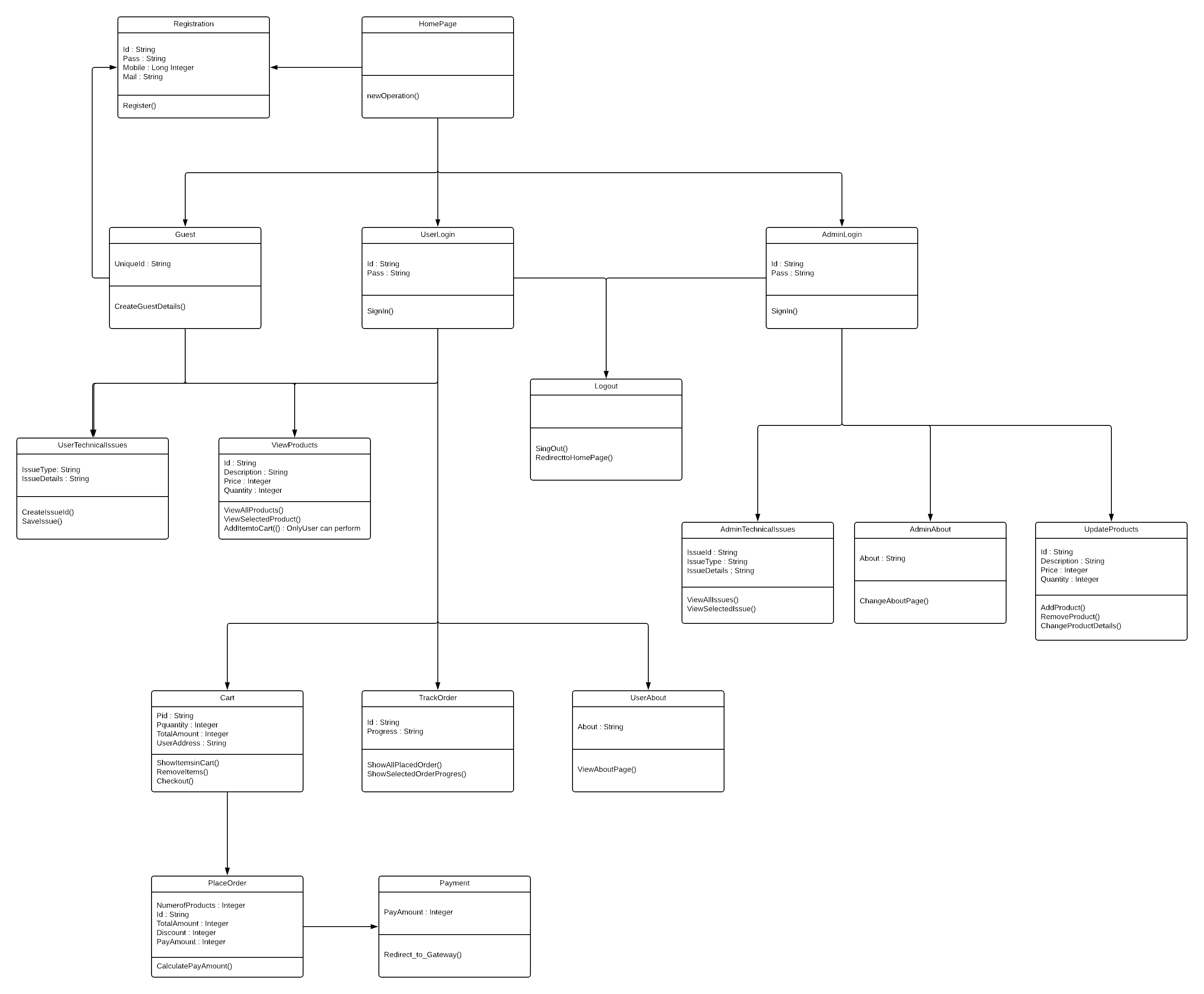
* + 1. **Introduction**: describes the document in brief, state the purpose of the document, define scope of the development project, definitions and abbreviations used in the document.
    2. **Conceptual Architecture/Architecture Diagram:** describes the overview of the modules or components we are using, Structure, relationships and dependencies of the components and issues of the user interface.
    3. **Logical Architecture:** describe the Logical Architecture Description of the components which includes the class, state and sequence diagram of each function.
    4. **Execution Architecture:** defines the runtime environment, processes, Deployment view.
    5. **Design Decisions and Trade-offs:** describes that what decision you choose while doing this project and reason why you chose that decision and if you have an alternative idea than what was the reason behind that you discard your alternatives over your decisions.
    6. **Pseudo code for components:** Describes the Pseudo code of the components.
    7. **Appendices:** describes other helpful information (if any).

1. **Conceptual Architecture/Architecture Diagram**
   1. **User interface issues**

* If the user is a person who has not operated a website will find difficulty to understand what is going on. For that kind of users we can provide helpful comments on the features that will help him to understand.
* If user has knowledge how to operate website but do not how to buy online product might find difficulties. These users have to learn about how payment gateway works. For these users we will provide contact information of company to ask about their doubts.

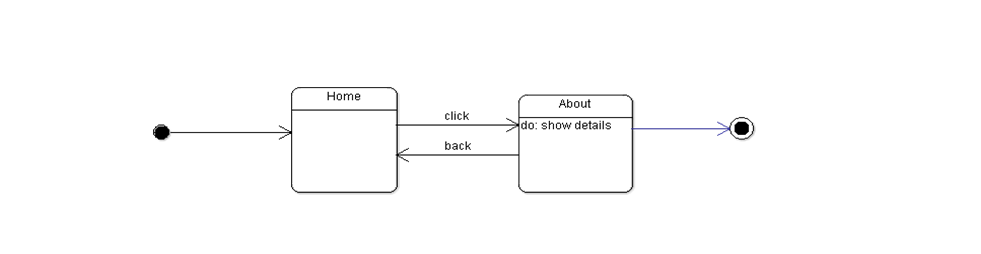
1. **Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)**

**3.1 Class diagram**

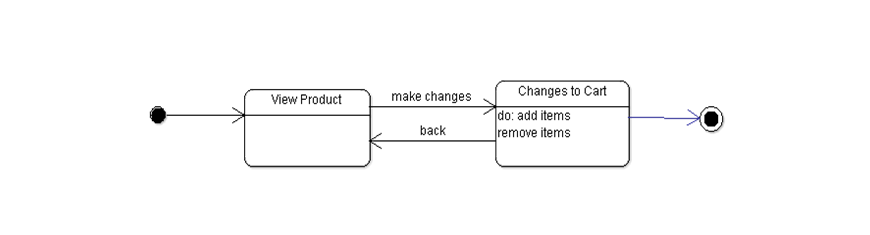
****

**3.2 State Diagram**

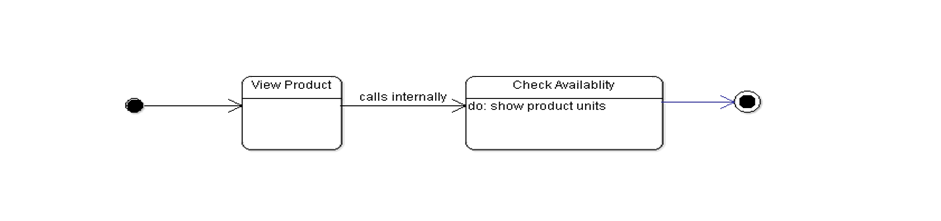
**3.2.1 About Page**

****

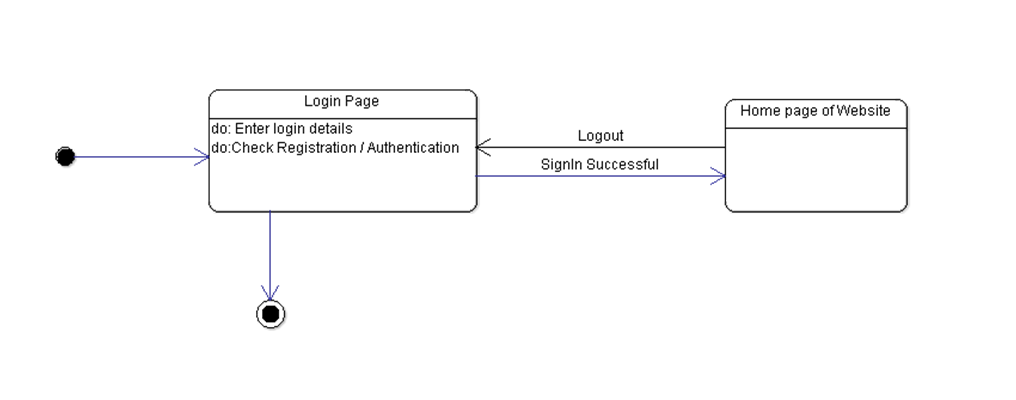
**3.2.2 Changes to the Cart**

****

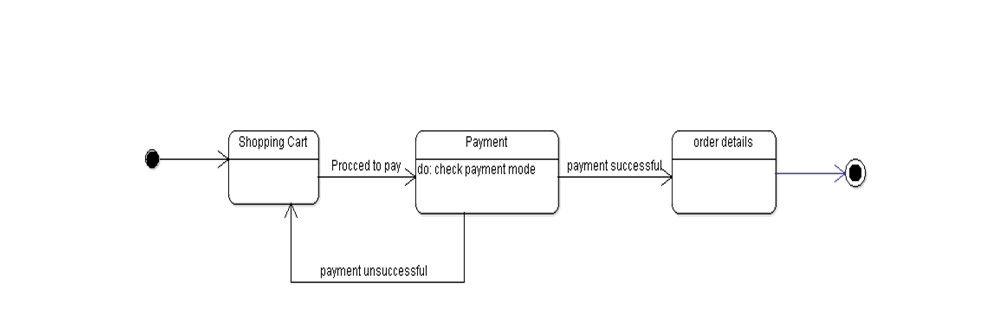
**3.2.3 Check Availability**

****

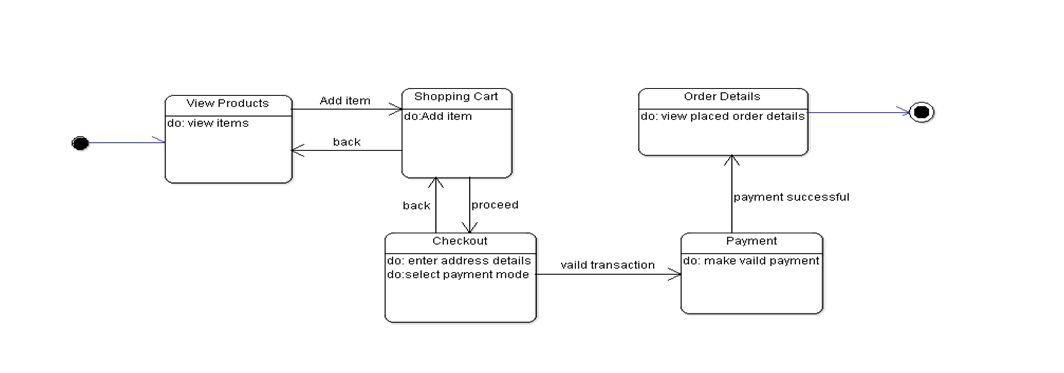
**3.2.4 Login Page**

****

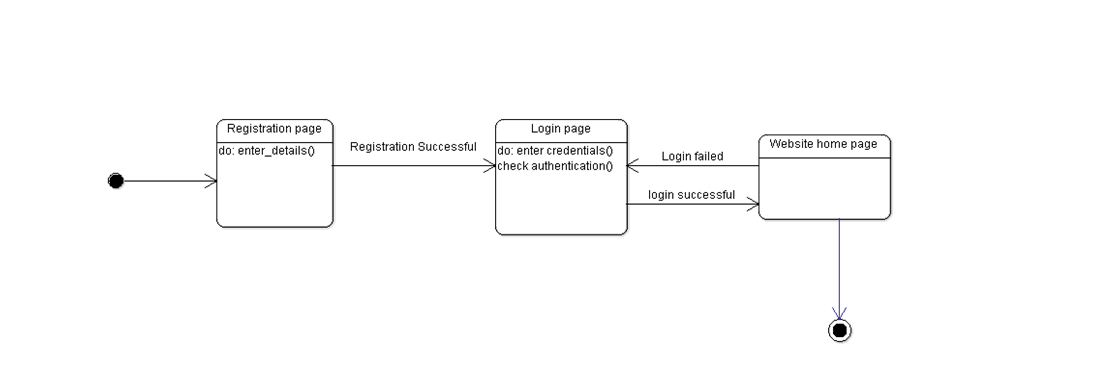
**3.2.5 Payment**

****

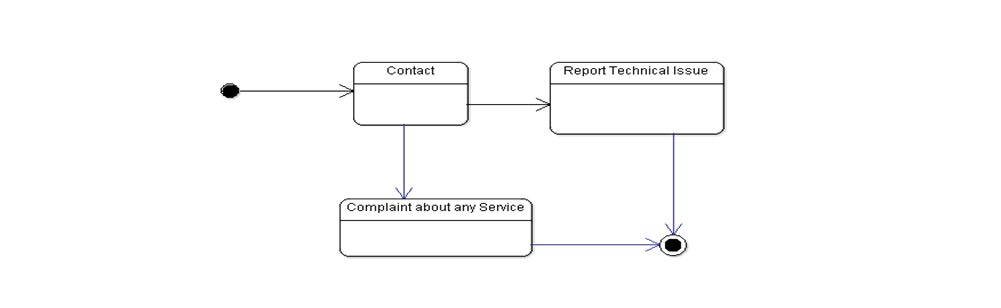
**3.2.6 Place Order**

****

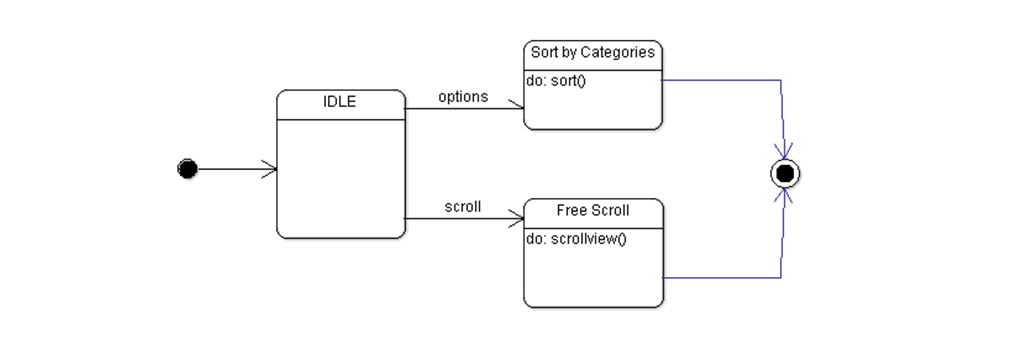
**3.2.7 Registration Page**

****

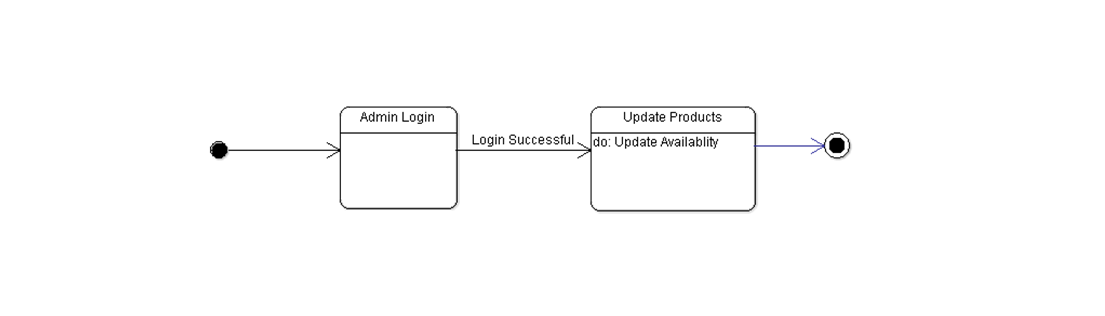
**3.2.8 Report Technical Issues**

****

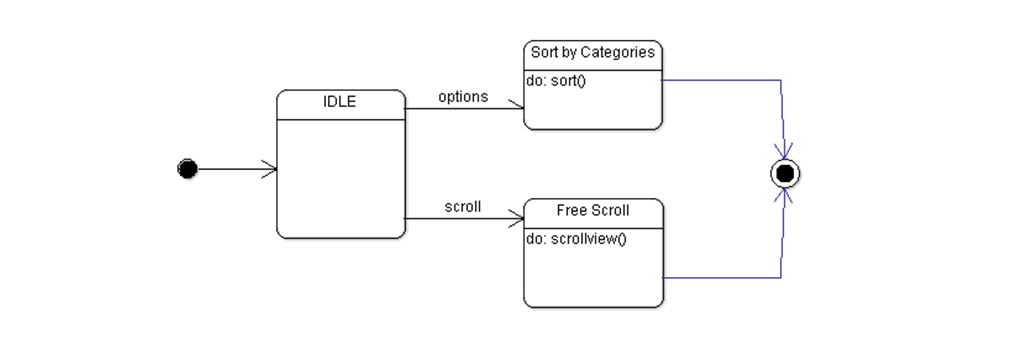
**3.2.9 View Updates**

****

**3.2.10 Update Products**

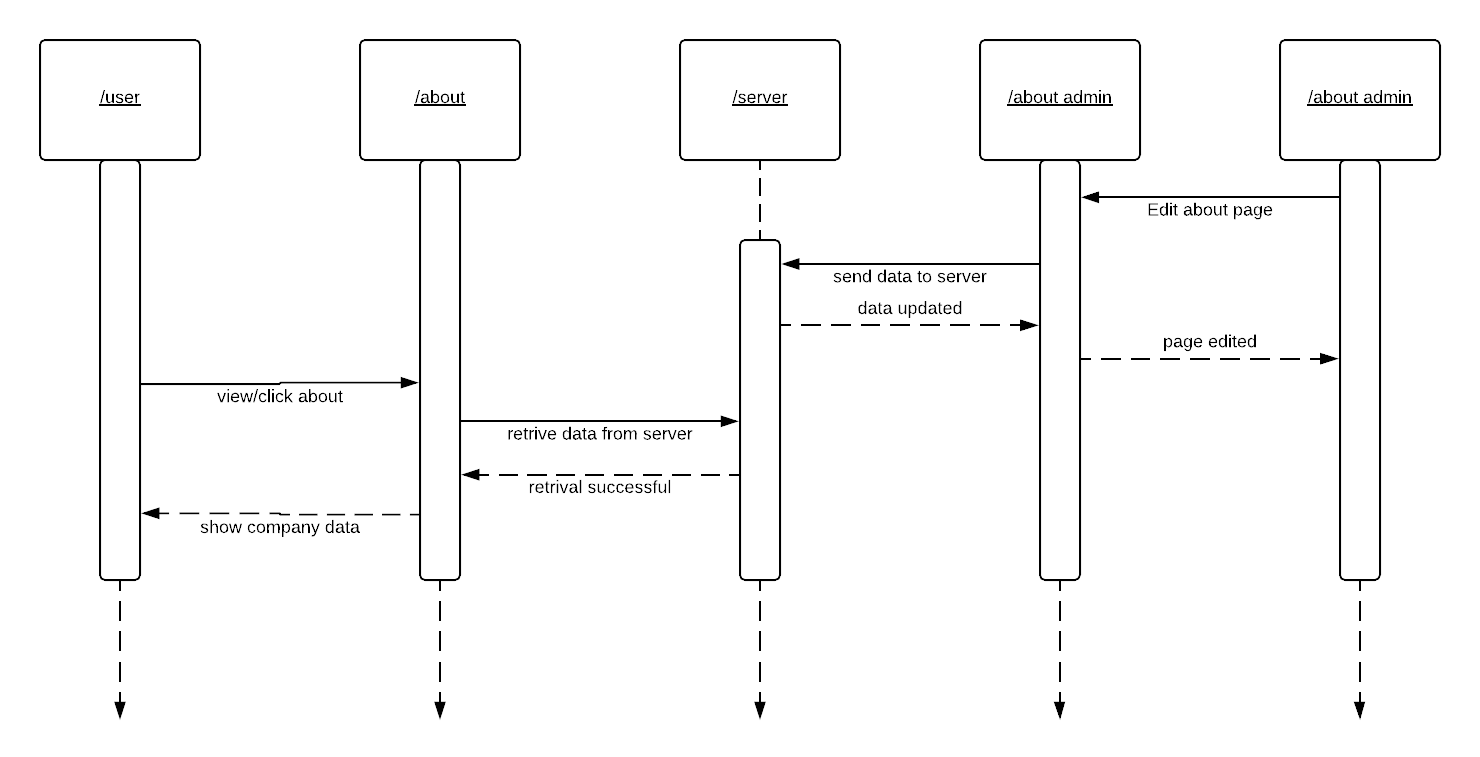
****

**3.2.11 View Products**

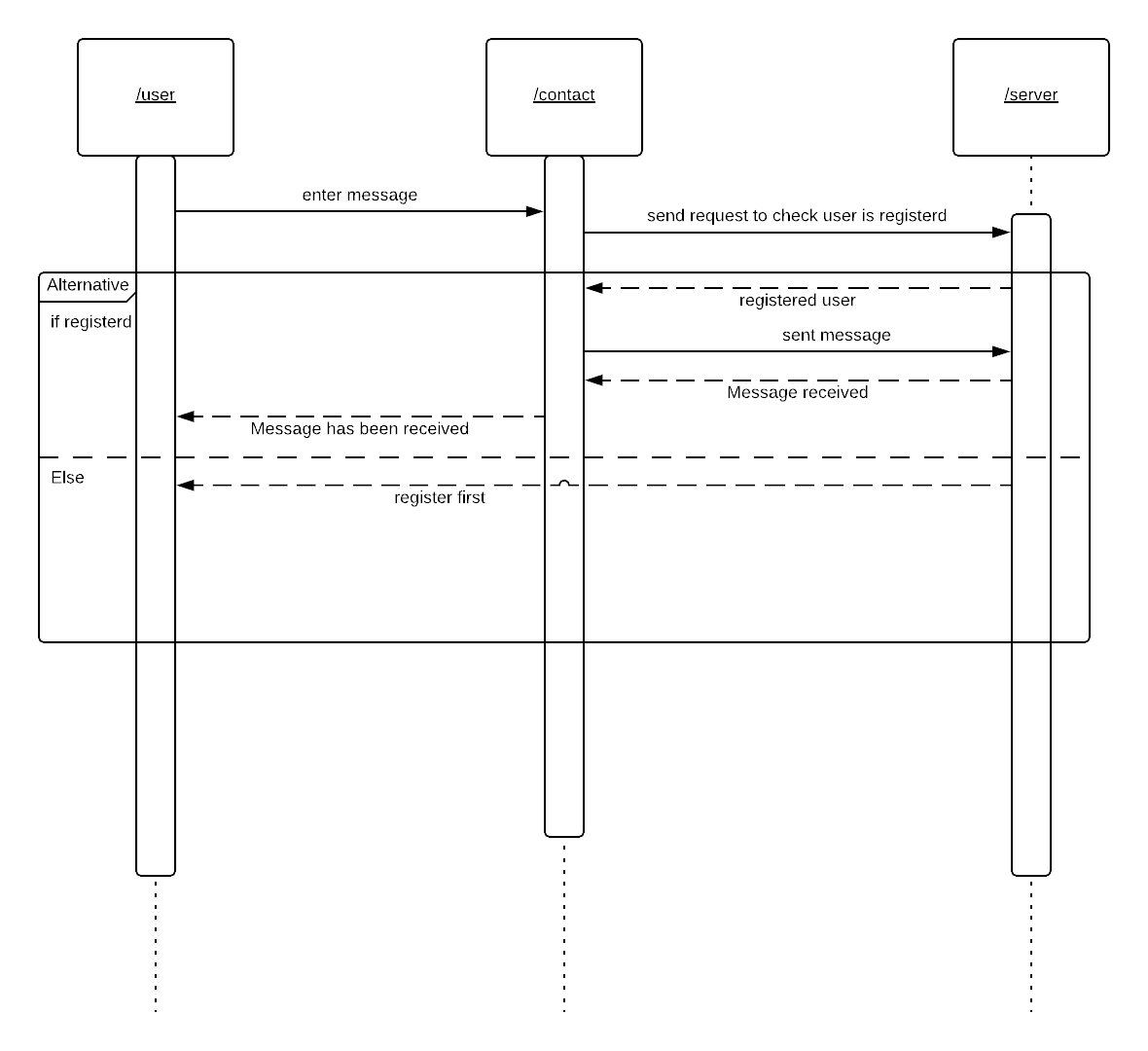
****

**3.3 Sequence Diagrams**

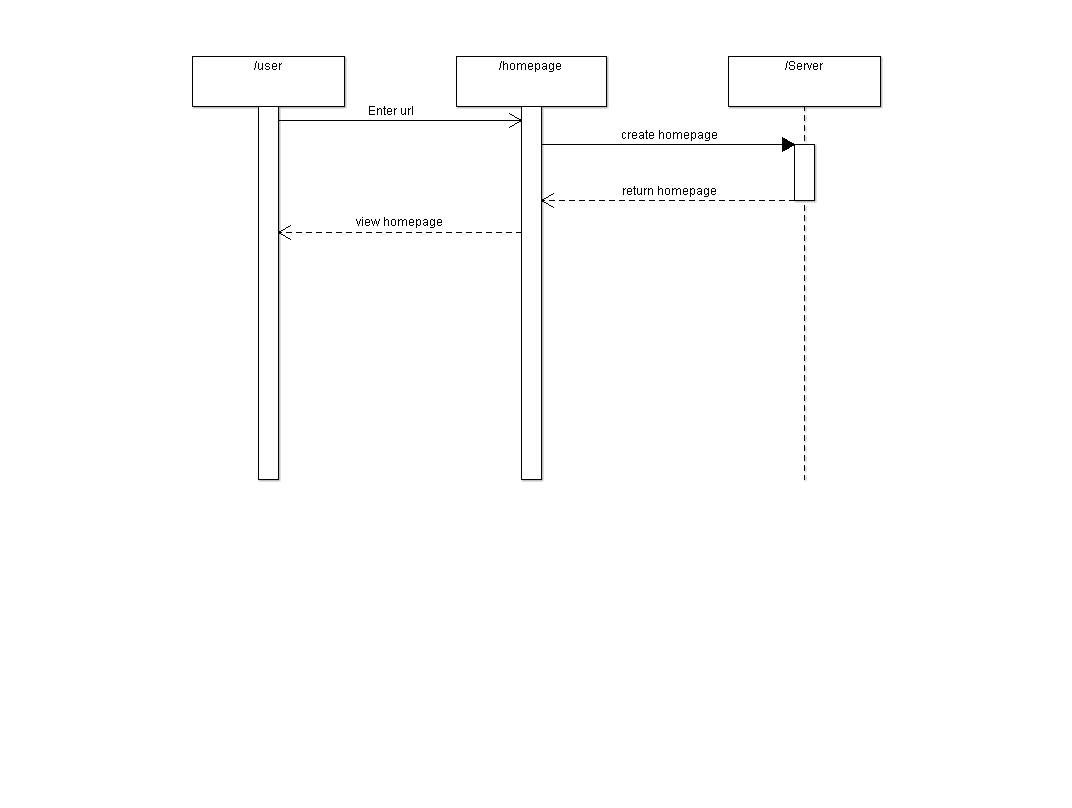
**3.3.1 About Page**

****

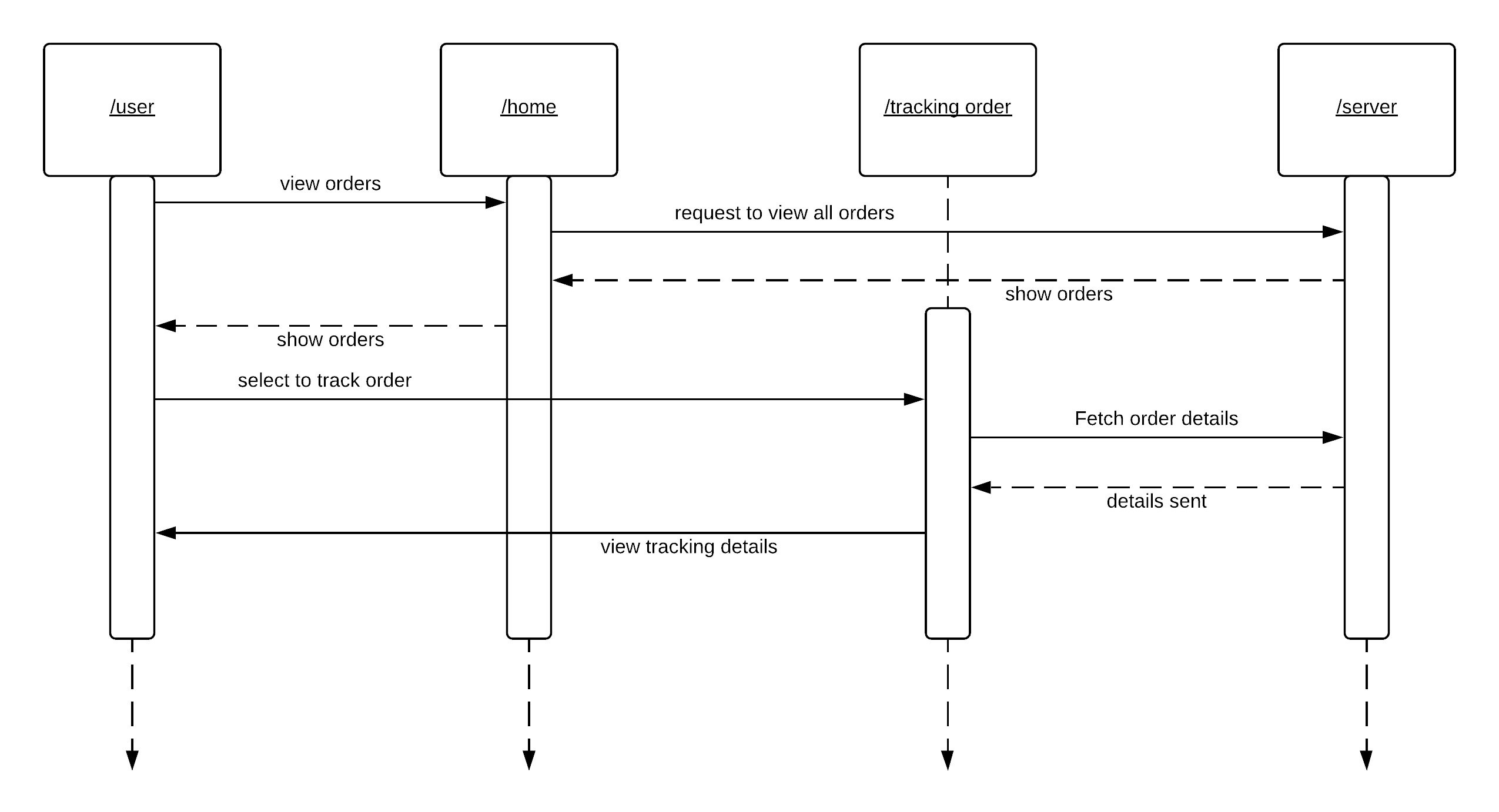
**3.3.2 Contact Page**



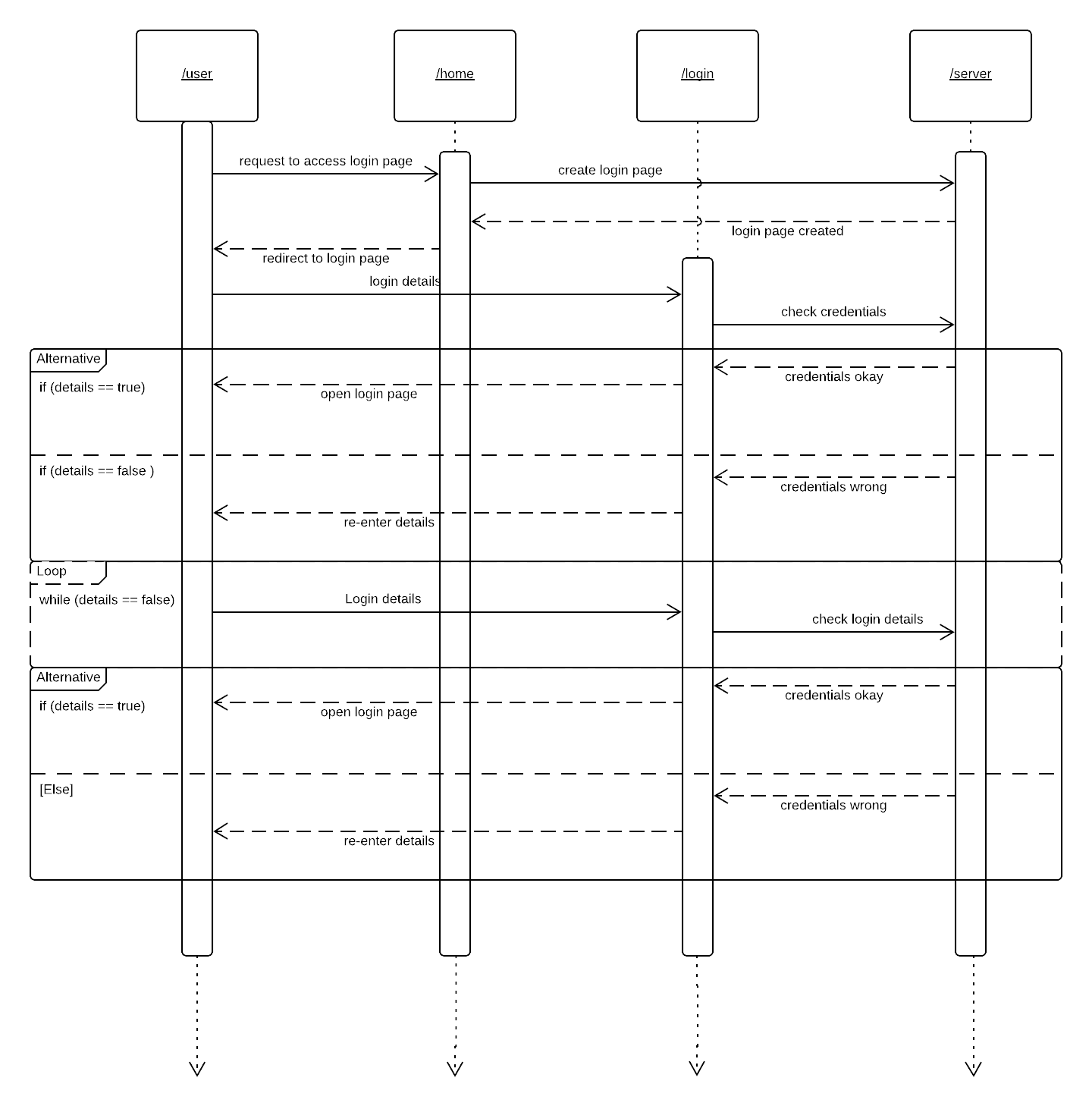
* + 1. **Homepage**



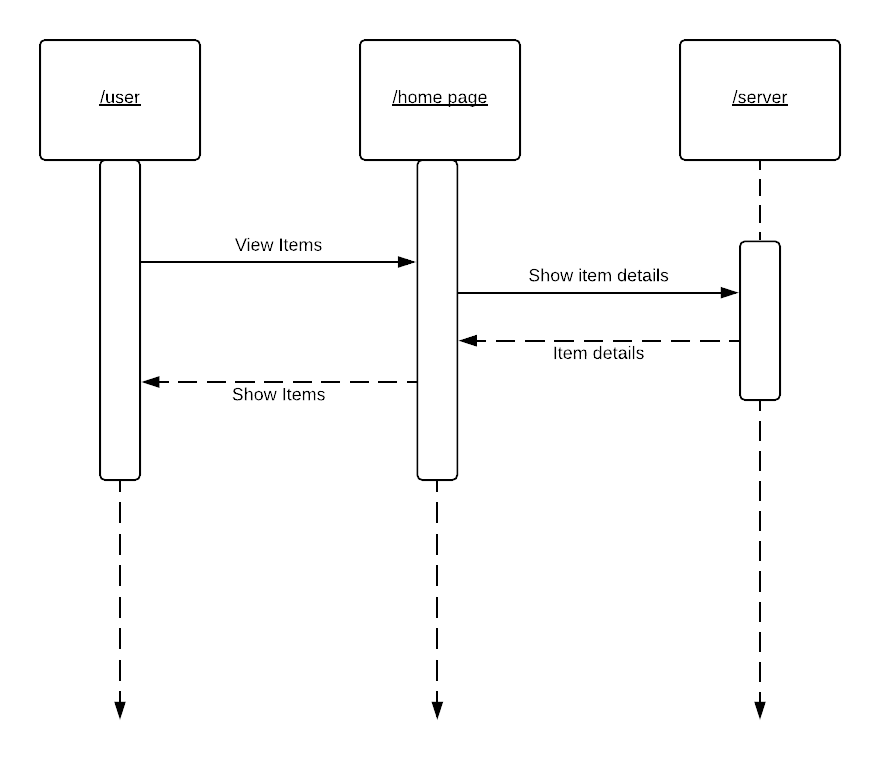
* + 1. **Tracking order**

****

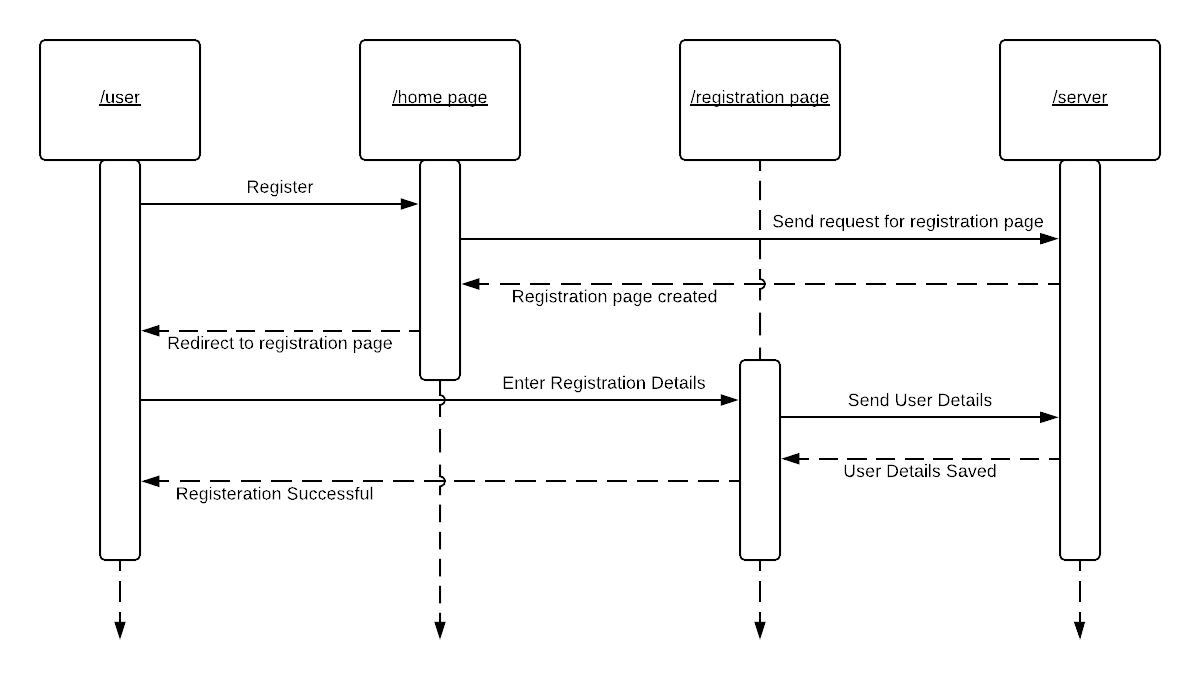
**3.3.5 Login**

****

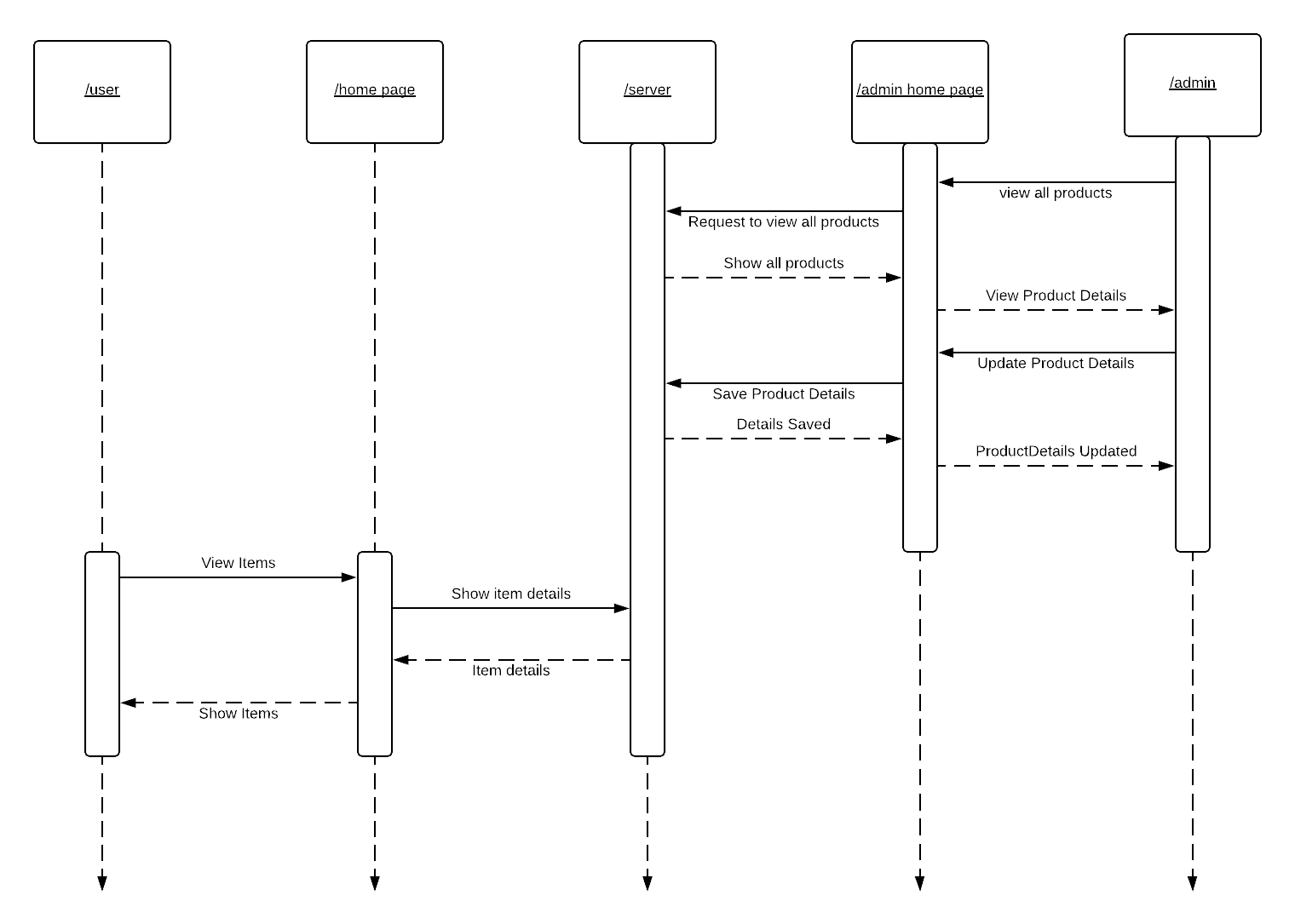
* + 1. **View Products**

****

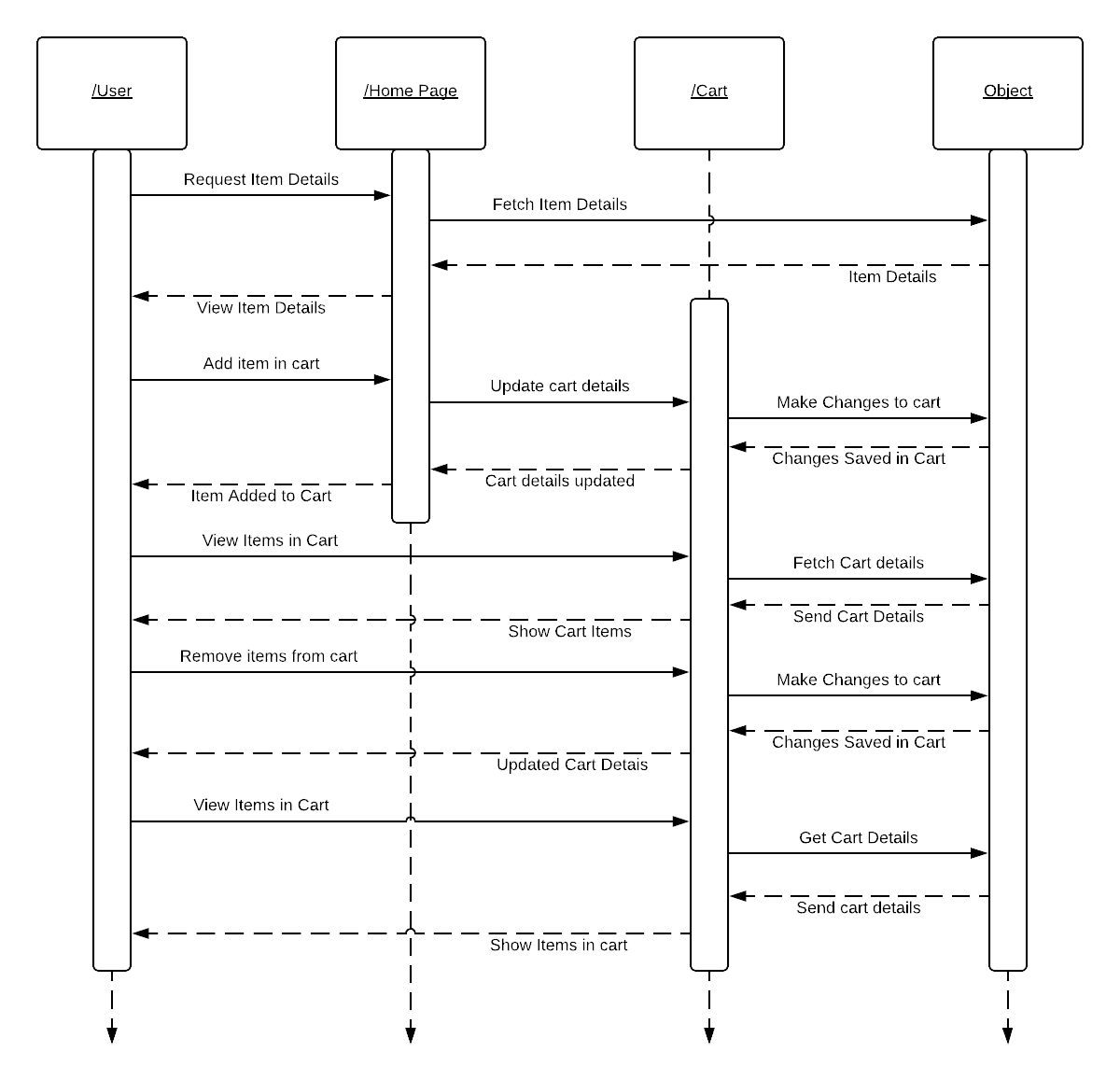
**3.3.7 Registration**

****

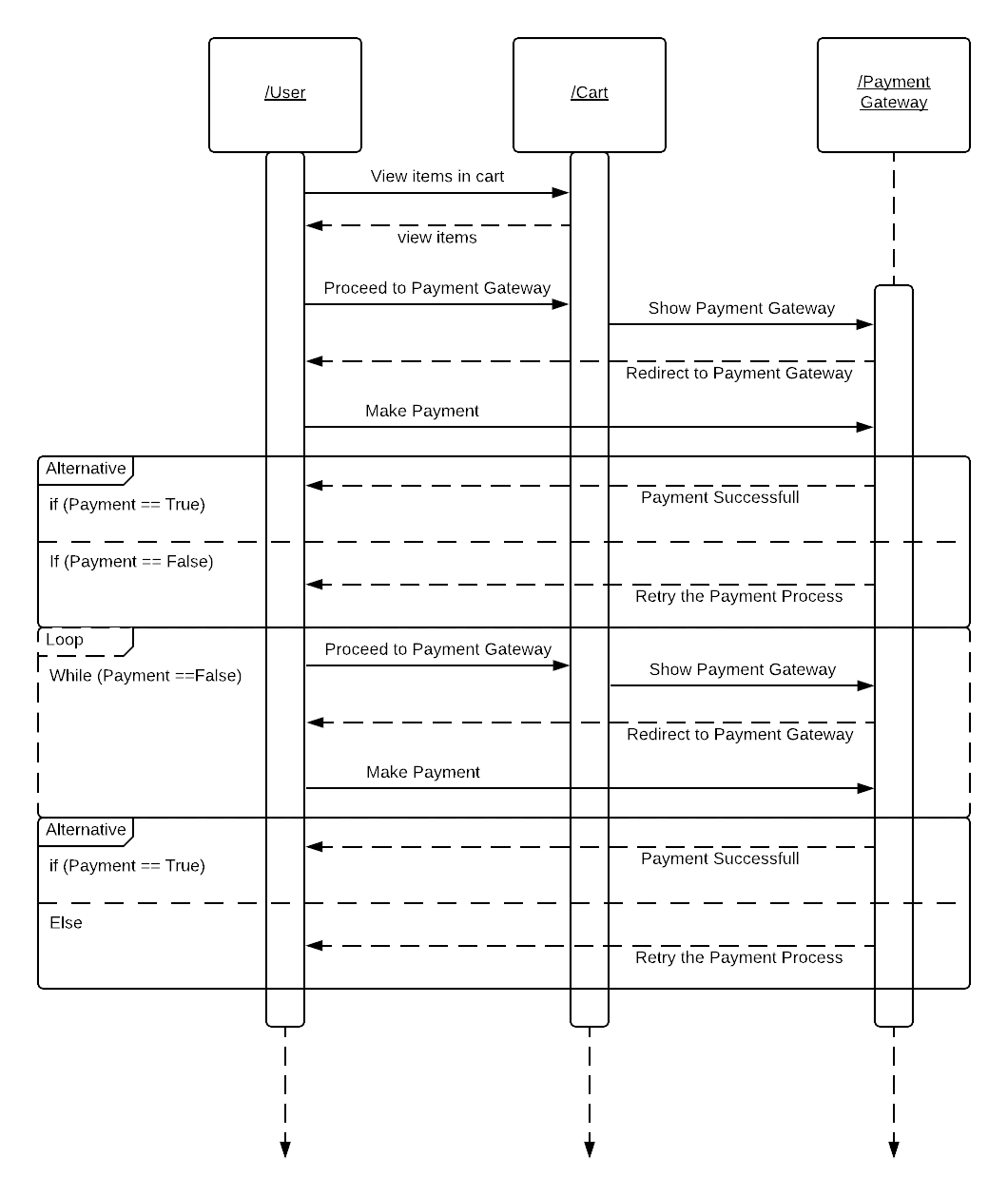
**3.3.8 Update Products**

****

**3.3.9 Changes to Cart**

****

**3.3.10 Place Order**

****

**3.1 Logical Architecture Description**

**3.1.1 Class Diagram Explanation**

Class diagram consists of the various classes which show the relation to other classes. Relation may be one-to-one or one-to-many. Our Class Diagram consists of 16 classes, which will completely explain how they are related to each other and how the functions are going to work. These classes are:

* UserLogin
* AdminLogin
* Guest
* Registration
* Logout
* Homepage
* ViewProducts
* UserTechnicalIssues
* AdminTechnicalIssues
* AdminAbout
* UpdateProducts
* Cart
* TrackOrder
* UserAbout
* PalceOrder
* Payment

These classes show the functionalities of both Admin and user. Most of the functions are fixed and handled by user and Admin classes. Diagram also explains the generalization of the Payment Gateway using various Payment platforms.

**Classes with explanations:**

1. **Class name : HomePage**

**Description:** This Class creates the home page, which is the initial page of the website and from here you can perform other functions provided.

1. **Class name : UserLogin**

**Description:** This class allows the user to login with his registered id and password.

**Input:** Id, Password

**Output:** Login Successful / Retry / Enter Valid Id and Password

**Method: SignIn()**

**Method Description:** This method accepts the id and password and verifies if it is correct. If it is correct, it will successfully log the user in. If id and password are not correct, it will tell you to try again.

1. **Class name : AdminLogin**

**Description:** This class allows the admin to login with his registered id and password.

**Input:** Id, Password

**Output:** Login Successful / Retry / Enter Valid Id and Password

**Method: SignIn()**

**Method Description:** This method accepts the id and password and verifies if it is correct. If it is correct, it will successfully log the admin in. If id and password are not correct, it will tell you to try again.

1. **Class name : Guest**

**Description:** This class allows the guest user to surf through our website.

**Method: CreateGuestDetails()**

**Method Description:** This method creates a unique id in order to keep track of all the things that the guest user does on our website, allow him to surf our website and view our products.

1. **Class name : Registration**

**Description:** This class allows a new user to register to our website and create an account for him.

**Input:** Id, Password, Mobile, Mail

**Output:** Registration Successful / Enter Valid Details

**Method: Register ()**

**Method Description:** This method takes all the details as input and saves them in the database and show a message ‘Registration Successful’. If the data entered is not according to the instructions, a message ‘Enter Valid Details’ will be shown and you will be asked to enter valid data/details.

1. **Class name : Logout**

**Description:** This class allows user/admin to logout of their accounts.

**Method: SignOut()**

**Method Description:** This method sign-outs user/admin of their accounts.

**Method: RedirecttoHomePage()**

**Method Description:** This method redirects user/admin to the home page after signing out of his/her account.

1. **Class name : AdminAbout**

**Description:** This class allowsadmin to change the contents of about page.

**Input:** About (String that saves all the details made to the about page)

**Output:** Changes Saved to the Page

**Method: ChangeAboutPage()**

**Method Description:** This method takes input from the admin as a string and saves the changes made by the admin to the about page, and then allows everyone to see those changes.

1. **Class name : AdminTechnicalIssues**

**Description:** This class allows admin to see all the issues faced by the users.

**Method: ViewAllIssues()**

**Method Description:** This method allows admin to see all the issues posted by the users.

**Method: ViewSelectedIssues()**

**Method Description:** This method allows admin to see the issues posted by user of a certain type or with a certain issueid.

1. **Class name : UpdateProducts**

**Description:** This class allows the admin to update the details of the products.

**Input:** Changed details like, ProductId, ProductDescription, ProductPrice, ProductQuantity ect.

**Output:** Changes made to the product.

**Method: AddProduct()**

**Method Description:** As the name specifies, this method allows the user to add a new product to the website.

**Method: RemoveProduct()**

**Method Description:** This method allows the admin to remove an existing product from the website.

**Method: ChangeProductDetails()**

**Method Description:** This method allows the admin to change any/all the details of products.

1. **Class name : UserAbout**

**Description:** This class allows the user to access the about page and get to know more about the organization and the owner.

**Method: ViewAboutPage()**

**Method Description:** This method allows the user to see the details provided by the admin about the organization and other details.

1. **Class name : UserTechnicalIssues**

**Description:** This class allows the user to post about the issues faced by the user, while using our website.

**Input:** IssueType, IssueDetails

**Output:** Your issue has been saved.

**Method: SaveIssue()**

**Method Description:** This method takes issueType and issueDetails as input from the user and then saves them.

**Method: CreateIssueId()**

**Method Description:** As the name specifies, this method creates a unique id for each issue that is being posted on the website.

1. **Class name : ViewProducts**

**Description:** This class allows the user to see the products available to buy.

**Method: ViewAllProducts()**

**Method Description:** This method allows the user to see all the products available.

**Method: ViewSelectedProduct()**

**Method Description:** Using this method user can see all the details of some selected products.

**Method: AddItemtoCart()**

**Method Description:** Using this method user can add any item he/she wishes to buy, to the cart.

1. **Class name : Cart**

**Description:** This class allows users to add, remove products in the cart, in order to buy them.

**Method: ShowItemsinCart()**

**Method Description:** This method allows the user to see all the items he/she has added in his/her cart.

**Method: RemoveItems()**

**Method Description:** This method allows the user to remove items from his/her cart.

**Method: Checkout ()**

**Method Description:** This method allows the user to place his/her order.

1. **Class name : PlaceOrder**

**Description:** This class allows the user to place his/her order.

**Method: CalculatePayAmount()**

**Method Description:** This method calculates the amount to be payed by the user.

1. **Class name : Payment**

**Description:** This class redirects you to the payment gateway. This class is automatically called by the PlaceOrder class.

**Method: Redirect\_to\_Gateway ()**

**Method Description:** This method helps to redirect user to the payment gateway linked to the website.

1. **Class name : TrackOrder**

**Description:** This class allows the user to track the progress of his order.

**Method: ShowAllPlacedOreders()**

**Method Description:** This method shows all the orders the user has placed.

**Method: ShowSelectedOrderProgress()**

**Method Description:** This method allows the user to select a particular order and know the current progress of that order.

**3.1.2 Sequence Diagram Explanation**

In Sequence Diagram, arrows are used to send messages to the other level. These messages are: Synchronous message – Solid line with a solid arrow head, Asynchronous message - Solid line with open arrow head and Response - Dotted line with open arrow head.

* + - 1. **About Page:**  Admin puts data on the about page about his company, and the users can see the data on about page in order to get to know about the organization.
      2. **Contact Page:** The user can put any complaints regarding any issues that he/she might have faced, while using the website.
      3. **Home page:** Home page is the main page of the website. The user can do various things on the home page. Like register, login, go to about page etc.
      4. **Tracking Order:** It lets the user to track his/her placed orders after logging in to his/her account.
      5. **Login:** It allows the user to login to his/her respective account, in order to access other functions on the website.
      6. **View Products:** It allows the user to view the products that are being provided by the organization.
      7. **Registration:** Unregistered users can register using this function, in order to place order, or use some other function of the website.
      8. **Update Products:** Only the admin can use this feature, in order to change details of the product, such as number of products available, add a new product, remove a product, change the description of a product etc.
      9. **Changes to Cart:** In order to use this feature you have to be a registered user. This feature allows the user to add items to his shopping cart, which he/she can buy thereafter.
      10. **Place Order:** This feature requires you to be a registered user. This feature lets you to place order, after selecting an appropriate payment option and then completing the process.

**3.1.3 State Diagram Explanation**

State Diagram explains the working of the functions in very effective way. Initial state is consisting of the black dot and final state is consisting of black dot surrounded by empty circle.

**3.1.3.1 About Page**

There is a separate about page for both user and admin. It allows user to view the details of the company like when the company was established or what is experience of the company. And It allows admin to update admin page with company information.

**3.1.3.2 Changes to the Cart**

Only User can perform this action. While viewing the product if user wants to buy some product then, he can add items to the cart or remove them as well.

**3.1.3.3 Check Availability**

This function is applicable for both admin as well as user. They can check the availability of the product. For users, while viewing the product there is an internal function of coding call the method to see the view count of the product.

For Admin, due to admin privileges admin can check availability on website as well as on server.

**3.1.3.4 Login Page**

**3.1.2.3.1 Login ()-**

**Input:** login credentials / username / password

**Output:** Login Successful or Login Failed

**Description:** Allows user to enter their credentials, which are being checked for authentication and if they are registered or not. This information is stored in the server. Login() access the user details from the server and once authentication is successful user can carry on the home page and have access to every feature of the website.

**3.1.3.5 Payment**

**3.1.3.5.1 Redirect\_to\_Gateway ()**

**Input:** Card / UPI Details

**Output:** payment Successful / Payment Unsuccessful

**Description:** It allows user to make payment for the products they are buying online. Payment can be done by any debit card or UPI methods. If there is any error in the payment then user is rolled back to the shopping cart page. If successful then order is confirmed, and user is taken to the order details page.

**3.1.3.6 Place Order**

Only users can place orders, while placing an order user is required to follow few steps

* View products
* Add product to cart
* Checkout (Add billing address details)
* Payment

After Payment order is confirmed and placed as well.

**3.1.3.7 Registration**

**3.1.3.7.1 Register ()**

**Input:** user credentials (must be new not stored in the server before)

**Output:** registration successful

**Description:** It allows user to register to the website and use all the features of the website.

**3.1.3.8 Report Technical Issue:**

It allows user to report the technical issues they facing during the session. It also allows user to complaint about anything suspicious (if found).

**3.1.3.9 Track Order:**

It allows user to track down there orders and see every recent update on their order.

**3.1.3.10 Update products:**

Only admin can use this feature to update the products on the website. Admin login is required to update the products.

**3.1.3.11 View Products:**

This is very simple process. User just has to enter the URL and then he will be redirected to the home page. To view the products he can freely scroll down or he can sort it by categories.

**4.0 Execution Architecture**

Runtime Environments are some main browsers used widely like Google chrome, Mozilla Firefox, internet explorer etc.

**4.1 Re - use And Relationships to Other Products**

NIL

**5.0 Design Decision and Trade-offs:**

We had lots of ideas of how we are going to explain our SDS. When we think about our project it goes very deep and we keep exploring new things. Few decisions which are trade-off during the making of the SDS:

* We have considered explaining the class diagram and its components just below the class diagram and same for state and sequence diagram also. But we find it more difficult because every function is coming very quickly and we haven’t completed the coding part. Second reason for not choosing this option was to go with the sequence of IEE SDS template format.
* We were explaining our most functions on the basis of user. It was so confusing while making the sequence diagram. Some functions are colliding with each other so we decided to explain function from admin side too. It helped us to get more accurate view on the project.

1. **Pseudo Code for Components:**

**7.0 Appendices:**

NIL