**Student ID:** **986826**

**Full Names:** **Dawit Araya Mebrahtu**

Software Engineering

(CS425)

(November 2018)

Professor: O. Kalu

Final Integration Exercise

1. The exam duration is 2 hours.
2. The exam is computer-based; so you may use a computer for both the coding and theory parts.
3. Make sure to switch-off your cell-phones or simply turn the ringer off.
4. **This exam is a copyrighted material and must not be copied or reproduced or transferred**.
5. You are expected to use an IDE or any Code Editor tool of your choice to implement your solutions for the questions in the Coding part. Upon completion, put your project(s), **(source code only)** in a single zip file named **FinalExam.zip**, including your completed/finished exam paper (i.e. this document – **in Microsoft Word or PDF format, only**), and submit to Sakai.

--------------------------------------------------------------------------------------------------------------------

Type your answers to the theory questions in the following pages.

--------------------------------------------------------------------------------------------------------------------

(CS425 - SWE)

(November 2018)

Final Examination (70 points)

**Part I – Theory (True/False, Short answers, Multiple-choice questions):** (16 points)

1. (2 points) **Science of Consciousness**:

Answer the following question; giving 2 sentences in support of your answer:

State two (2) reasons why our twice daily practice of Transcendental Meditation can be considered to be an Agile technique for improving our brain performance?

**Answer:**

**Yes. Because practicing TM twice daily enhances our thinking ability. And also, scientific research has shown that twice daily practice of TM, results in us having higher brain coherence (e.g. Dr. Fred Travis’s research).**

1. (2 points) Answer the following questions with True or False.
   1. (1 point) The technique of “mocking” an object or component or service implies creating a “fake” one to serve as a placeholder for the actual object or component or service, for the purpose of implementing Unit tests, in isolation. True or False?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**TRUE**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. (1 point) In the RUP process model, during the Architecture Design activity, the class coded below will be globally visible to all other classes across the system. True or False?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**FALSE**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**package** edu.mum.cs.cs425.prodmgmt.global.publicc;

**class** GlobalPublicUtils {

**public** **static** **final** **long** ***MAX\_INVENTORY\_COUNT*** = 10000L;

// **TODO**

}

1. (12 points) Give short answers to the following questions.
   1. (2 points) Explain the difference between the following 2 JPA annotations. (You may give examples to illustrate your answer).
      1. @Column(name="ProductNumber", **nullable=false**) private long productNumber;
      2. **@NotNull** private long productNumber;

**@Column is the JPA way of declaring a column there by indicating databse schema details, with column name ProductNumber and not null constraint in this case. Whereas @NotNull is a bean validation annotation and is used for validating input.**

* 1. (2 points) With respect to relationship between two classes/entities, what do we mean by the term, Dependency. Give an example using code snippets and/or diagram.

**Dependency between two classes/entities indicates that one class is dependent on the second class in order to perform some of its tasks. Dependency is different from other kinds of relationships in that the dependent class is not required to keep reference of the class in which it depends on as its own attribute.**

**Public class MathDependecny{**

**Public static void main(String[] args){**

**System.out.println(Math.sqrt(16));**

**}**

**}**

**The above code snippet shows that class MathDependecny has a dependency on the class Math.**

* 1. (2 points) With respect to relationship between classes/entities, what do we mean by the term, Composition? And how does it differ from an Aggregation? Give examples to illustrate your answer.

**Composition is a restricted form of aggregation in which two classes are highly dependent on each other. It represents part-of relationship. For example, an order is a whole and line items are parts. If an order is deleted then all corresponding line items for that order should be deleted. So Composition has a stronger relationship.**

**Aggregation is a special form of association. It represents has-a relationship and is a unidirectional association. In aggregation both classes can survive individually. For example, a line item is a whole and product is a part. If a line item is deleted then corresponding product need not be deleted. So aggregation has weaker relationship.**

* 1. (6 points) For the MUMScheduling project, given below is Use-case specification for when a faculty who owns a Specialization track, reviews pending track enrollment requests.

**Use-case:** Faculty Specialization Track Owner Reviews Pending Track Enrollment Requests.

**Brief Description**:

This use case allows a faculty owner of a specialization track to review pending specialization track requests from students and to mark them as approved or rejected.

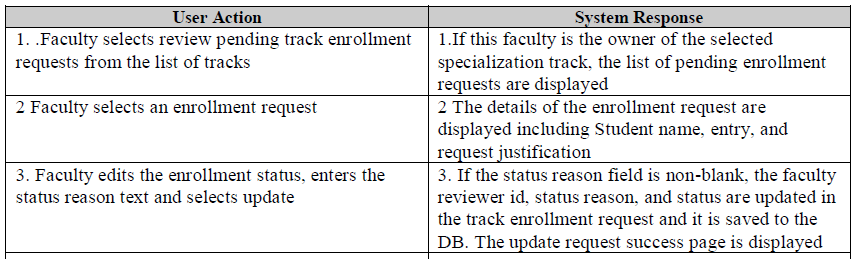
**Actors**: Faculty

**Preconditions**:

Faculty is logged in and at the view specialization track page.

**Flow of Events**:

**Basic Flow**



**Post-Conditions:**

For the success basic flow the track enrollment request is saved to the DB with updates.

Business Rules:

1. Only the faculty owner of the specialization track can review track enrollment requests.
2. The status reason entry is text to explain the reason for accepting or rejecting an enrollment request. It must be non-blank when setting the status to accepted or rejected.
3. The status of track enrollment requests can be pending, accepted, or rejected.

Using ATDD, write a User Story for the above use-case, representing the basic success scenario. In your user story, indicate (in brackets, at the end of each statement) what section of the RUP use-case description, as given above, corresponds to each of your User story statement.

**Answer:**

**User story:** Reviews Pending Track Enrollment Requests.

* 1. As a Faculty
  2. I want to review pending track enrollment requests.
  3. So that I can mark the requests as approved or rejected.

**Scenario:**

Reviews Pending Track Enrollment Requests from faculty Enrollment Requests page.

* 1. Given I am viewing Enrollment Requests
  2. When I click approved request
  3. Then the request is marked as approved and doesn’t appear in the Enrollment Requests page.
  4. Finally, the request is saved to the DB with updates.

|  |  |
| --- | --- |
| 1. **When** faculty selects review pending track enrollment request from the list of tracks | 1. If the faculty is the owner of the selected specialization track, **then** the list of pending enrollment request are displayed. |
| **2.When** faculty selects an enrollment request. | 2. **Then** the details of the enrollment request are displayed including Student name, entry, and request justification. |
| 3. Faculty edits the enrollment status, enters the status reason text and selects update | 3. If the status reason field is non-blank, **then** the faculty reviewer id, status, reason, and status are updated in the track enrollment request and it is saved to the DB. The update request success page is displayed. |

**Part II – Software Engineering Problem-solving, Coding skills:** (54 points)

**Note:** *For these questions, for each of your solution, you are expected to take screenshot(s) of your result(s), save it into a .png or .jpg image file and include these in the FinalExam.zip file you submit.*

1. (12 points) **Implementing Unit Testing using the JUnit framework**:

Implement code for a component named, MyArrayUtils, as a Java command-line (console) application. In your component (class), implement a method named, **hasMultipleMaximum(…)**, which takes as input, any array of integers and it returns true, if the maximum integer value in the array, occurs more than once, otherwise it returns false. For example, when given an input, a\_in = [-6, 2, 5, 6, -6, 5, 6], your hasMultipleMaximum method should return the value, true. And when given an input, a\_in = [-6, 2, 5, -6, 5, 6], the method should return the value, false.

Also, when hasMultipleMaximum (...) is called and the input argument passed is null or an empty array, it should return, false.

Using JUnit, implement unit tests for your MyArrayUtils component and its hasMultipleMaximum () method; covering the following 3 test-cases:

1. When the input is a legit integer array, such as [-6, 2, 5, 6, -6, 5, 6] or [-6, 2, 5, -6, 5, 6].

2. When the input is a null.

3. When the input is an empty array.

Create a JUnit TestSuite containing your 3 test-cases defined above. And execute your TestSuite and take a screenshot of your result, as displayed by your IDE.

1. (42 points) **Implementing an Enterprise Web Application**

A local SuperStore, named WallyMarty, has hired you to design and develop a Customer Relationship Management (CRM) system for them, which they will be using to collect, maintain and manage data about their customers. They want you to implement a basic web application for this purpose. Especially important to the store is, their select group of PrimeCustomers, who they offer a special 10% discount on every purchase that customers belonging in this group, make at the store.

A PrimeCustomer is a customer who is of age, 40 or older.

Here are the attributes for the Customer entity:

Customer: customerId:long, customerNumber:string, name:string, contactPhoneNumber:string, dateOfBirth:date

For this question, you are expected to do the following:

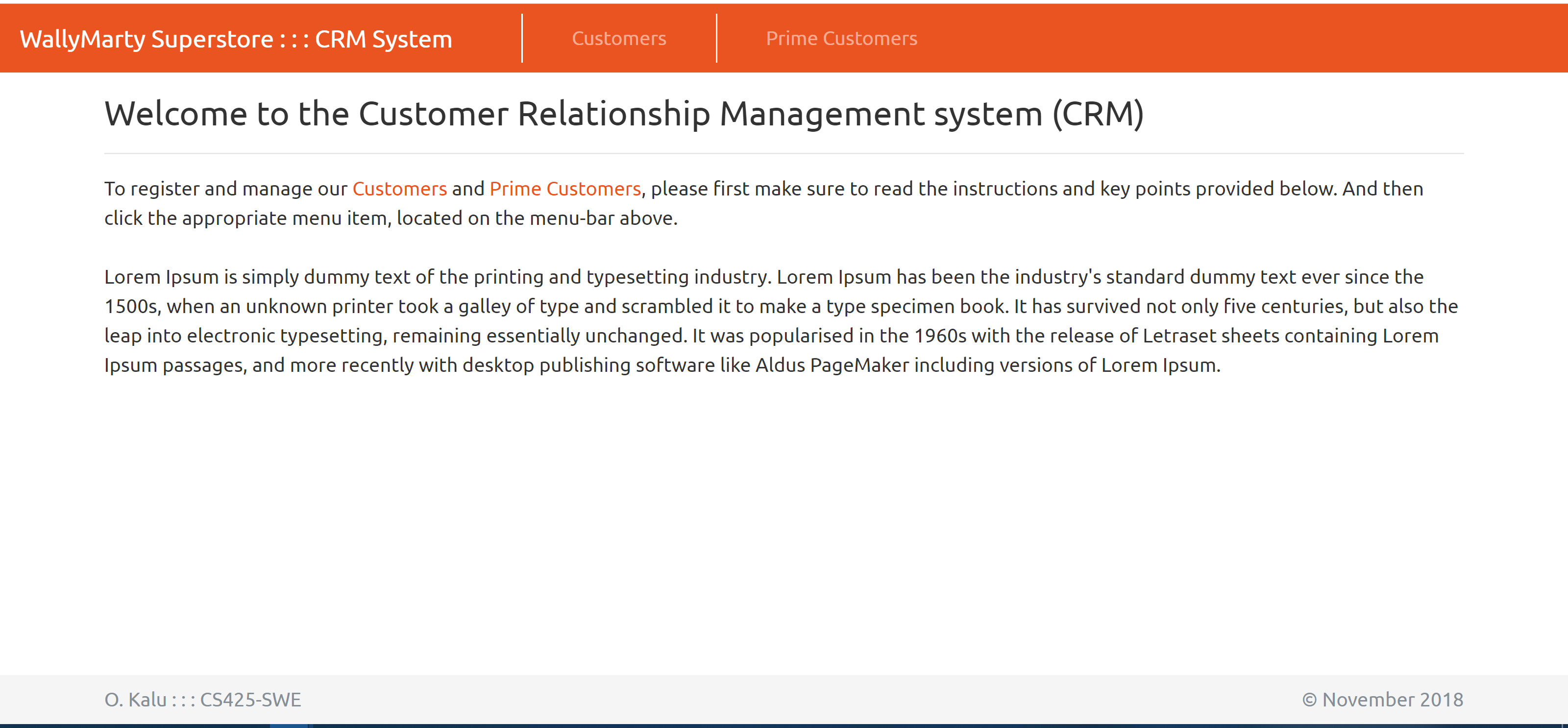
1. Using the set of tools, technologies and frameworks which you have learnt about in this CS425 course, including Spring Boot, Spring Web MVC, Spring Data JPA, etc., (or some other Enterprise Web application development platform/tool(s) that you prefer), implement a working web application for WallyMarty SuperStore. You may use any database of your choice.

You are expected to implement only the following use-cases:

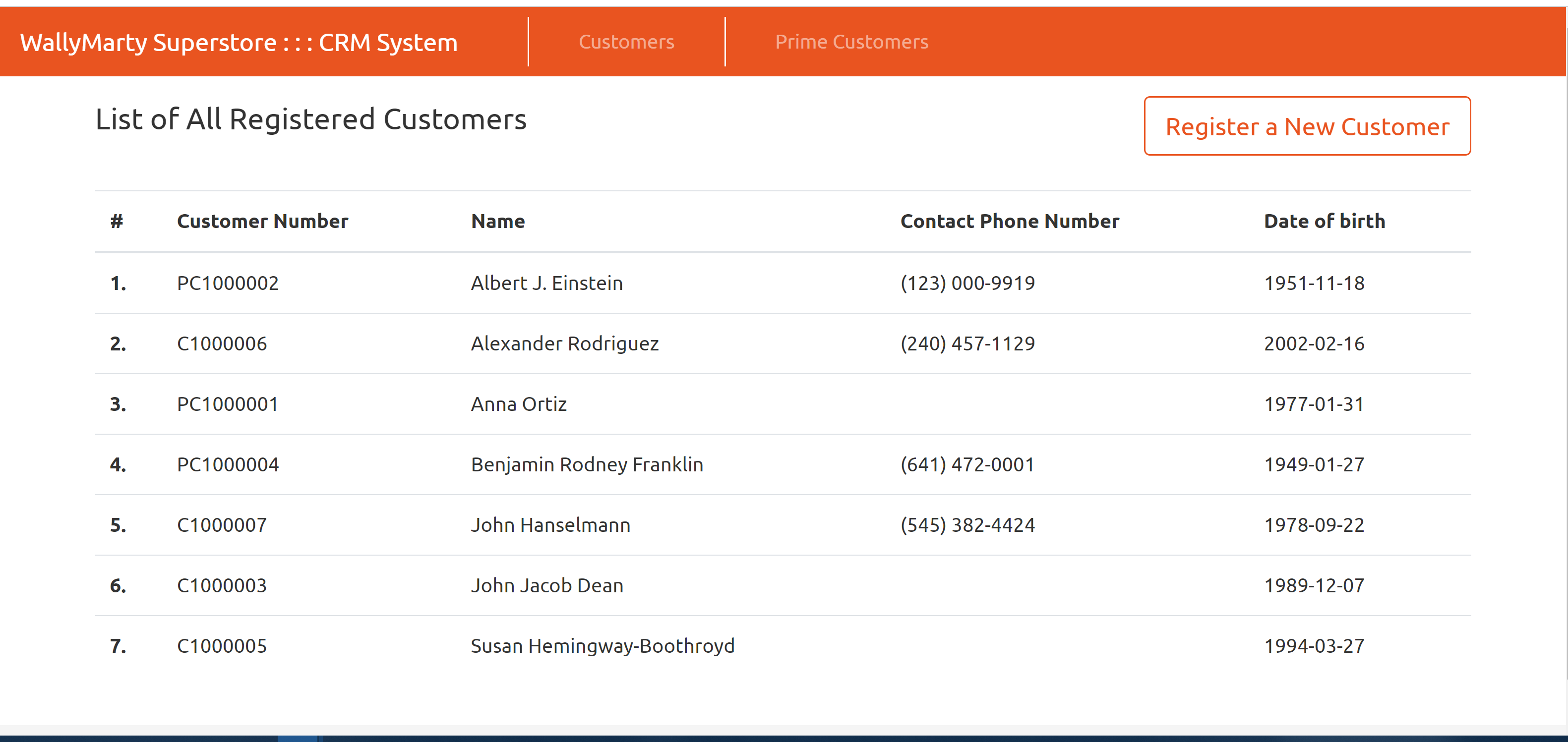
1. Display the application’s Homepage.
2. Display list of Customers (Allows the store manager to view a list of all the Customers registered in the system). The store requires this list to be displayed sorted in ascending order of the Customers’ names (see sample screen below).
3. Register a new Customer (Allows the store manager to add a new Customer into the system).
4. Display list of PrimeCustomers (Allows the store manager to view a list of all the PrimeCustomers in the system). The store requires this list to be displayed sorted in ascending order of the Customers’ dates of birth (see sample screen below).

Shown below are sample User Interfaces for the use-cases. Note: Your own UI design does NOT necessarily have to look exactly like these samples. But your UIs should contain all the necessary data and data fields, as shown.

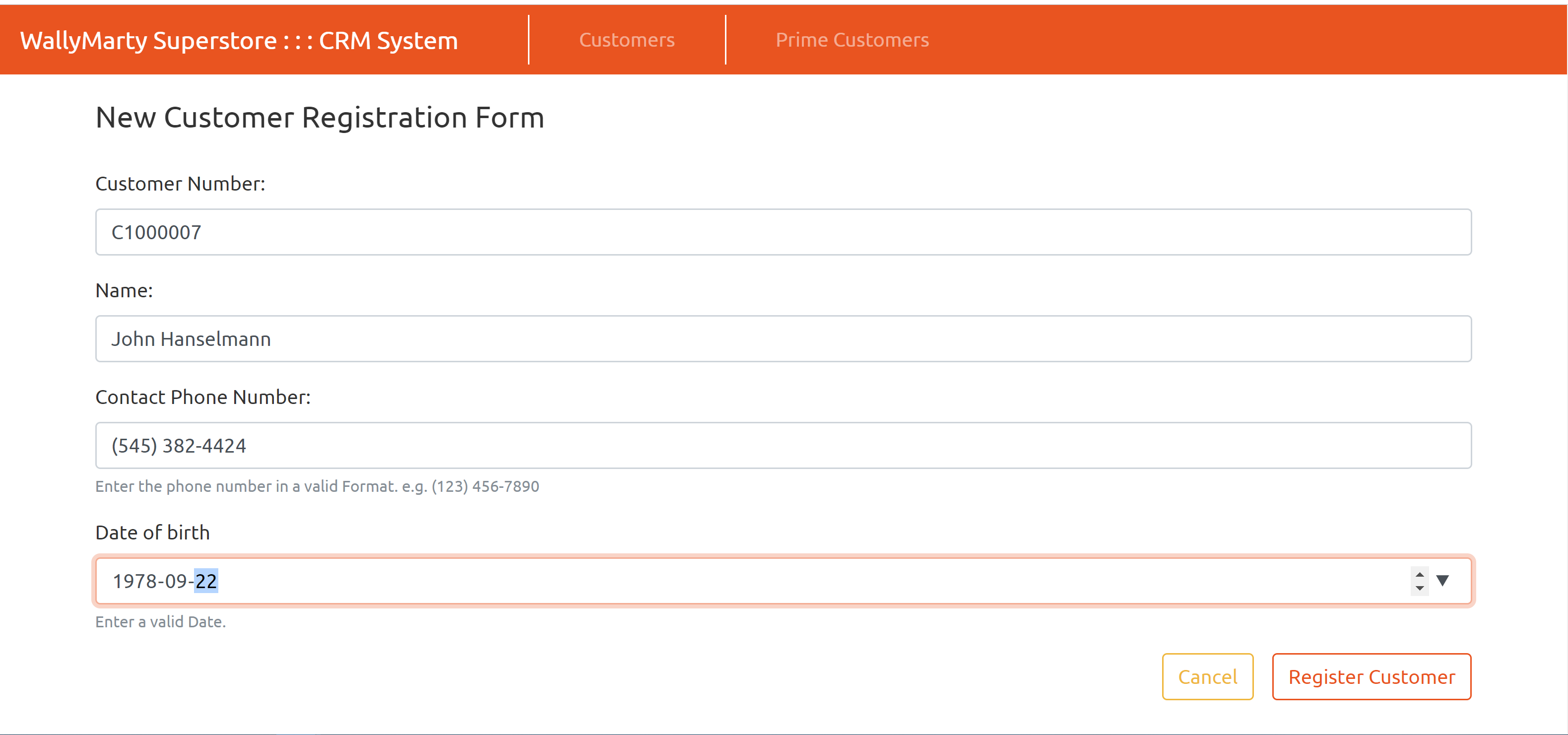
**Homepage:**



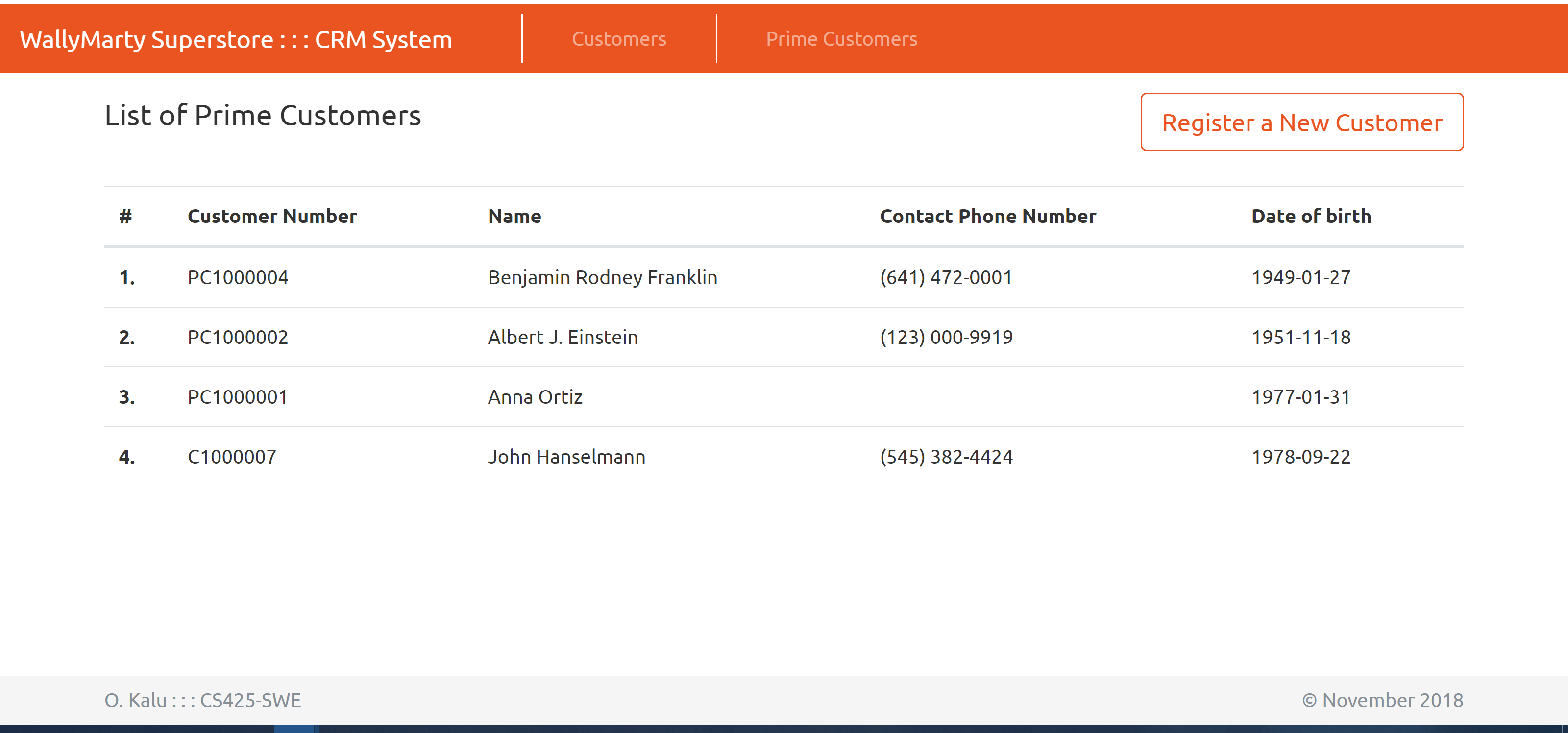
List of all Registered Customers:



Register a new Customer:



List of Prime Customers:



**//-- The End --//**