A MINI PROJECT REPORT

**On**

**GLACO**

**(GLA Canteen Online Mgmt.)**

**Submitted by**

**SHUBHAM KUMAR**

**Roll No: 151500522**

**SUSHIL SINGH**

**Roll No: 151200563**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



**GLA University**

**Mathura- 281406, INDIA**

**December, 2017**



**Declaration**

We hereby declare that the work which is being presented in the Mini Project “ GLACO (GLA Canteen Online Mgmt.)“ in partial fulfillment of the requirements for Mini-Project LAB, is an authentic record of our own work carried under the supervision of **MR. NEERAJ VARSHNEY, Assistant Professor, GLA University, Mathura**.

**Signature of Candidate 1**

**Name of Candidate: SHUBHAM KUMAR**

**Roll. No. : 151500522**

**Course: B.TECH (CSE)**

**Year: 3rd**

**Semester: V**

**Signature of Candidate 2**

**Name of Candidate: SUSHIL SINGH**

**Roll. No. : 151200563**

**Course: B.TECH (CSE)**

**Year: 3rd**

**Semester: V**

**CERTIFICATE**

This is to certify that the project entitled **“*GLACO*”** carried out in Mini Project Lab is a bonafide work done by ***Shubham Kumar 151500522*** *and* ***Sushil Singh 151500563*** and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

**Signature of Supervisor:**

**Name of Supervisor: Neeraj Varshney**

**Date:**

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to **Dr. (Prof). Anand Singh Jalal, Head of Dept., Department of CEA** for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe special debt of gratitude to **MR. Neeraj Varshney**, **Assistant Professor,** Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

SHUBHAM KUMAR

SUSHIL SINGH

**Abstract**

A web based tool that can generate short summary of large texts or books. Inputs will be in form of text. It is a process of shortening the text document with software, in order to create a summary with the major points of the original text Document, automatic data summarization is a part of machine learning and data mining. The main idea of summarization is to find a subset of data which contains the “information” of the entire set and key phrase extraction as well.

POS tagging and classification- A POS tagger is a piece of software that reads the text in some language and assigns parts of speech to each word(and other Token), such as noun, pronoun, verb, adjective, adverb, preposition etc.

**Table of Contents**

|  |  |
| --- | --- |
| Declaration | ii |
| Certificate | iii |
| Acknowledgments | iv |
| Abstract | v |
| Table of Contents | vi |
| 1. **Introduction** | **1** |
| 1.1 Motivation and Overview ………………………………........ | 1 |
| 1.2 Objective .…………………………………………………... | 1 |
| 1. **Software Requirement Analysis** | **2** |
| 2.1 Define the problem ……………………………………….. | 2 |
| 2.2 Define the modules and their functionalities (SRS) ……… | 3 |
| 1. **Software Design** | **6** |
| 3.1 Use Case Diagram………….………………………………… | 7 |
| * 1. DFD Diagram……………………..………………………… | 8 |
| * 1. Activity Diagram……………………………………………. | 9 |
| 3.4 Sequence Diagram………………………………………….. | 14 |
| 3.5 ER Diagram…………………………………………………. | 18 |
| 1. **Testing** | **19** |
| 4.1 Introduction to system testing………………………………. | 19 |
| 4.2 Type of Testing………..…………………………………….. | 20 |
| 1. **Implementation and User Interface** | **23** |
| 5.1 Implementation Interface……………………………........ | 23 |
| 5.2 User interface………………………………....................... | 24 |
| 5.3 Output Screen……………………………………………… | 25 |
| **References/Bibliography** | **31** |
| 1. **Appendices** |  |

**Chapter 1**

**Introduction**

* 1. **Motivation and Overview**

It has always been a question to the computer engineers that how can we make our machine understand the human language which is high level language and cannot be easily interpreted by the machine but in this era of machines, it is now possible for the machines to understand the human language, but still there is a lot left to be done. Machines still cannot process our language very well but we’re still putting efforts to make it fully accurate.

**1.2 Objective**

Online ordering system that we are proposing here, greatly simplifies the ordering process for both the students and the faculties of GLA University. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Users can choose one or more items to place an order which will land in the Cart. Users can view all the order details in the cart before checking out. At the end, Users gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Canteen Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

**1.2.1 Purpose**

This document presents a detailed explanation of the objectives, features, user interface and application of GLA Canteen Management System in real life. It will also describe how the system will perform and under which it must operate. In this document it will be also shown user interface. Both the consumers and the developers of the system can benefit from this document.

* + 1. **Scope**

This system will help to manage and run the restaurant business systematically. In this management system, we will provide an app that can be used by the users to order food. Users can also give feedback through this app. So that owner of the canteen can evaluate the whole system. Users can also make payment through Paytm wallet or GLA wallet which will be integrated with the management software. All the information about daily expenses and profit will be saved in the system. Also the required information about orders items will be saved in the system which can be only accessed by the system admin.

**Chapter 2**

**Software Requirement Analysis**

**2.1 Problem**

Our Project is based on automation of GLA Canteen which allows users to pay canteen bills using online wallet and request for delivery of available canteen items in the menu.

The project consists of android app and a web services .The android app will be used by the students and faculties of university whereas the web service will be used by the canteen management.

**2.2   Modules**

The entire project consists no. of modules on Android App and Web Server side.

**2.3 Modules Functionalities**

The android app will contains certain modules. The modules are as follows:

1. **Login and Registration**

Registration: The user can register through android app by filling up the registration form which consists of fields (Username, Roll no, Faculty Id, GLA mail id, Mobile no, Password etc.). When user submits the form, an OTP is sent to the given mail Id for verification of the user, as user complete the verification they will be able to use the services.

Login: The user can login to the app by providing their mail-Id, Roll no and password. In case user forget the password then he/she can also reset it through registered mail id.

**2.  Canteen Menu**

After login user will be able to see the menu items available at the canteen. They can add the items in the cart.

**3.  Cart**

The items added from the menu can be seen in the cart, user can add/remove the item from the cart. After adding the items in the cart user will verify the details before going to the payment gateway.

**4.  Wallet**

The app will have two type of wallets, which are

* Third party wallet like paytm/mobiwik etc (any one will be used).
* G-wallet: User can recharge this wallet using its canteen-id from canteen. All the refunds (in case) will be added in this wallet.

**5.  Delivery**

1. The user can opt  for delivery after paying for the required items.
2. The user will get a delivery confirmation from the canteen management  whether the order is accepted or not. If the order is accepted the items will be delivered to the user at the given address else the total paid amount will be added/transferred to the G-wallet.
3. The user have to provide an alternate delivery address in case he/she will not be able to receive due to any reasons.
4. In case the person at given alternate address is not able to receive the order then the total paid amount exclusive of delivery charges will be added\* to the G-wallet.
5. If the same incident is done by user twice, his/her account will be added to the blacklist and they will not be able to order in future with that account but can be able to pay the canteen bills directly only.
6. The user will confirm the delivery after receiving the order in the presence of delivery-boy.
7. Some delivery charges might be applicable according to the amount of order.

**6.  User Profile**

In this, user will be able to manage their profile details and see payment and order history.

**7.  Notifications**

User will get notifications for its payment, orders and delivery etc.

The web service will contains certain modules. The modules are as follows:

1. **Login and Registration**

Registration: The canteen management can register through web service by filling up the registration form.  When user submits the form, an OTP is sent to the given mail Id for verification, on completion the verification they will be able to use the services.

Login: The management can login by providing their mail-Id and password. In case user forgets the password then he/she can also reset it through registered mail id.

**2.  Order**

The canteen management can view the orders placed by the users and can accept/reject the order for delivery. On accepting/rejecting the delivery request a notification will be sent to the user. When the user confirms the completion of delivery, the management will be able to see that the order is completed.

**3.  Management Profile**

In this, management will be able to manage their profile details and see payment and order history.

**4.  Search User Details**

The management will be able to see all the orders and payments done by the users by entering their Id.

**Chapter 3**

**System Design**

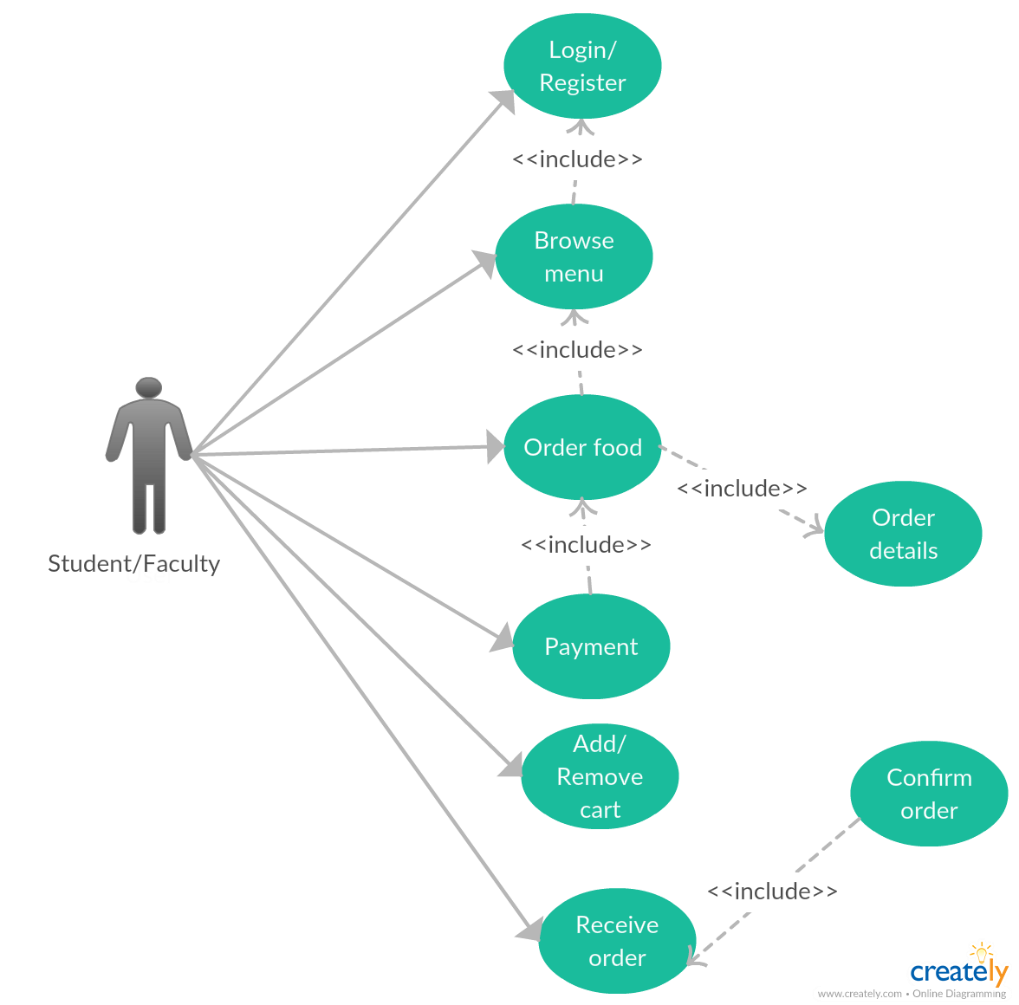
**UML Diagram**

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs . you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system.

1. Use case Diagram
2. Data flow diagram
3. Activity Diagram
4. Sequence  Diagram
5. ER Diagram

**3.1 Usecase diagram**

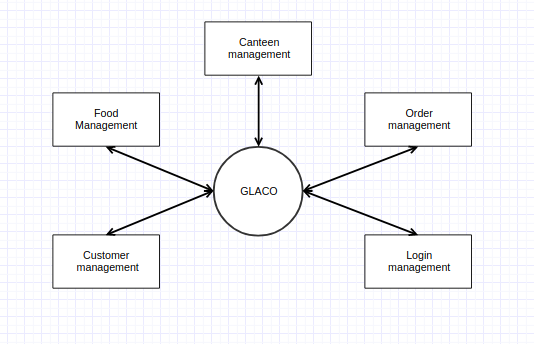
A use case diagram in the Unified Modelling Language(UML) is a type of behavioural diagram defined by and created from a use-case analysis .Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases),and any dependencies between those use cases.



**Fig 1**

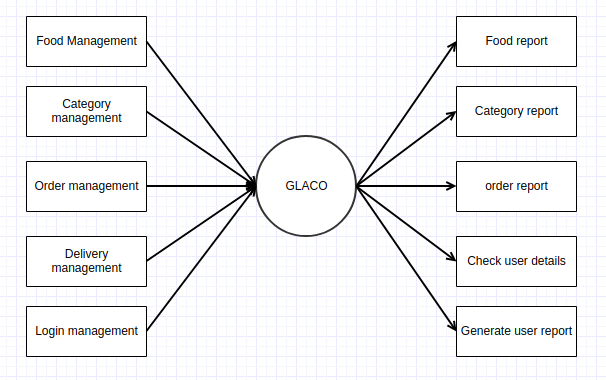
## 3.2 Data Flow Diagram

## 3.2.1. Level 0 DFD:



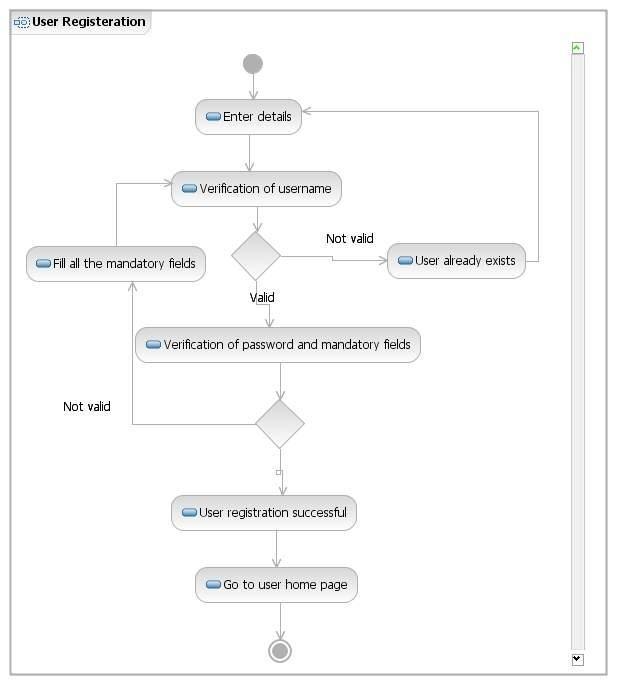
**FIG 2**

## 3.2.2 Level 1 DFD:



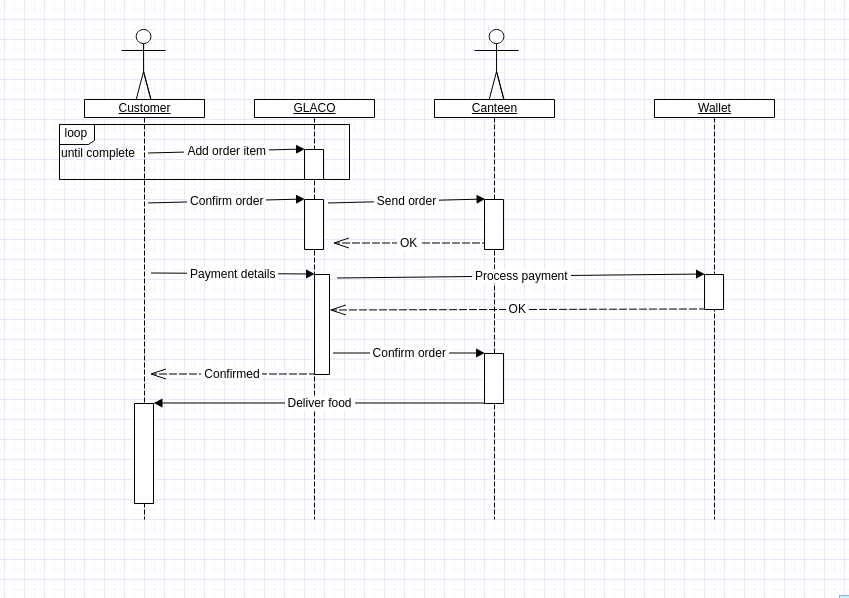
**Fig 3**

**3.3. Activity diagram**



**Fig 4**

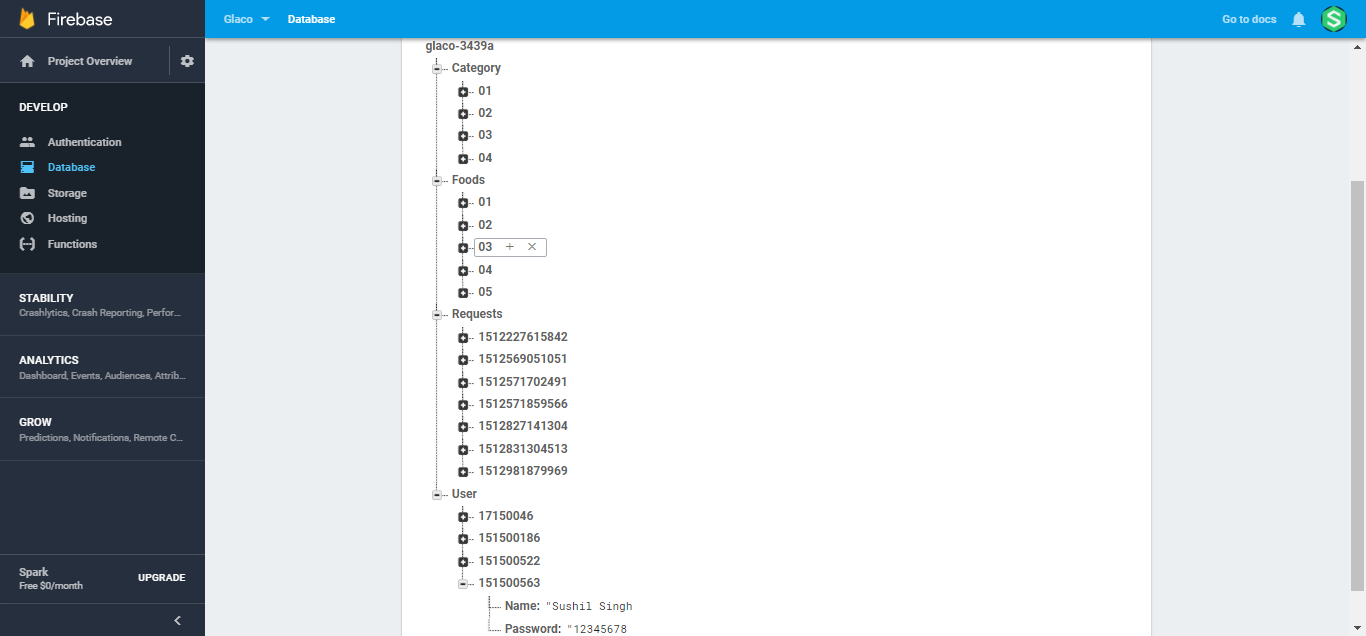
**3.4 Sequence Diagram**



**Fig 5**

**3.5 Database design**

**3.5.1 Database tables**

****

**Fig 6**

**3.5.2 ER diagram**

## https://lh3.googleusercontent.com/pvgpWNG3SRfPFhZhpxyno_ca_-GdWbAnCDa4HUz0H7lTtzeIMBzCEvD8BBfB4-nykNnZ0GPKvWuL8lXrBc0vjDDgArTY8Y2d1KVd8v_LW81OuKduD22GilD2MP4b-W_XsdeFk6Y

**Fig 7**

**Chapter 4**

**Testing**

**4.1 INTRODUCTION TO SYSTEM TESTING:**

 The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**4.2 TYPES OF TESTING**:

**4.2.1 Unit testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**4.2.2 Integration testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program.  Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at   exposing the problems that arise from the combination of components.

**4.2.3 Functional test:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input              :  identified classes of valid input must be accepted.

Invalid Input            : identified classes of invalid input must be rejected.

Functions                 : identified functions must be exercised.

Output                     : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**4.2.4 System Test**:

 System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**4.2.5 White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**4.2.6 Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**4.3 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**4.3.1 Test strategy and approach**

 Field testing will be performed manually and functional tests will be written in detail.

**4.3.2 Test objectives**

∙         All field entries must work properly.

∙         Pages must be activated from the identified link.

∙         The entry screen, messages and responses must not be delayed.

**4.3.3 Features to be tested**

∙         Verify that the entries are of the correct format

∙         No duplicate entries should be allowed

∙        All links should take the user to the correct page.

**4.4 Integration Testing:**

 Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

 The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

**4.5 Acceptance Testing:**

 User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

**Chapter 5**

**Implementation and User Interface**

**5.1 Implementation Hardware/Software** **Interface:**

This section lists the minimum hardware and software requirements needed to develop the system efficiently.

**5.1.1 Hardware** **Interface:**

* Pentium Processor and above
* 50 GB of free hard-drive space
* 4 GB of RAM
* The application requires a smartphone running on Android OS v4.0+ (recommended).

**5.1.2 Software** **Interface:**

* Operating System            : Windows
* Technology : Node JS
* Web Technologies : Html, JavaScript, CSS, Bootstrap
* Web Server : Node JS
* Database : MySQL
* Browser                                        : Mozilla ,chrome

**5.2 User interface**

**5.2.1 Hardware** **Interface:**

* Pentium Processor and above
* 50 MB of free hard-drive space
* 128 MB of RAM
* The application requires a smartphone running on Android OS v4.0+ (recommended).

**5.2.2 Software** **Interface:**

* Operating System            : Windows / Linux/ OSX
* Browser                                        :   Mozilla, chrome etc.

**5.3 Output Screen**



                                                   FIG-8:  Home page of GLACO

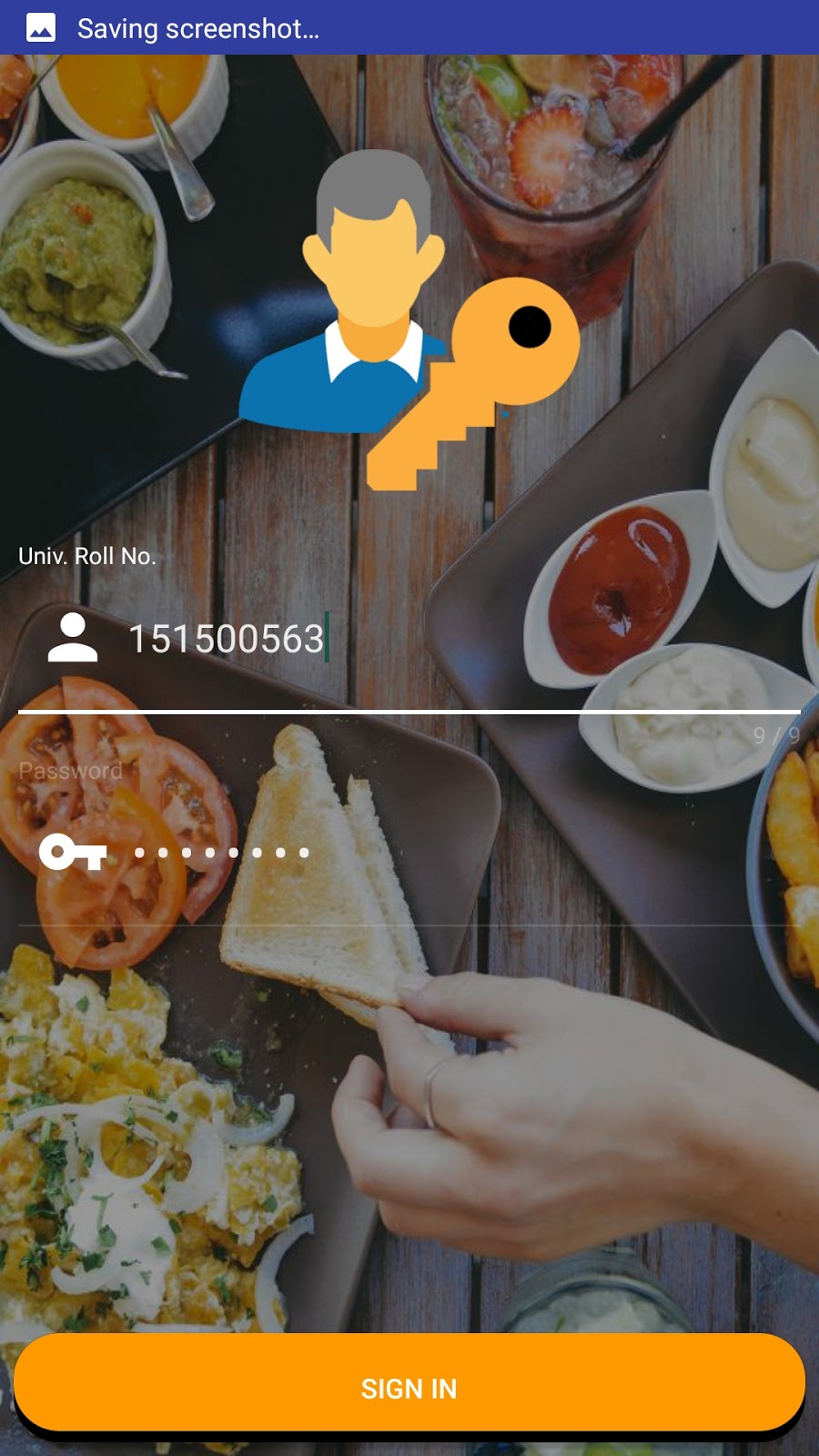


              Fig 9:  Login Page

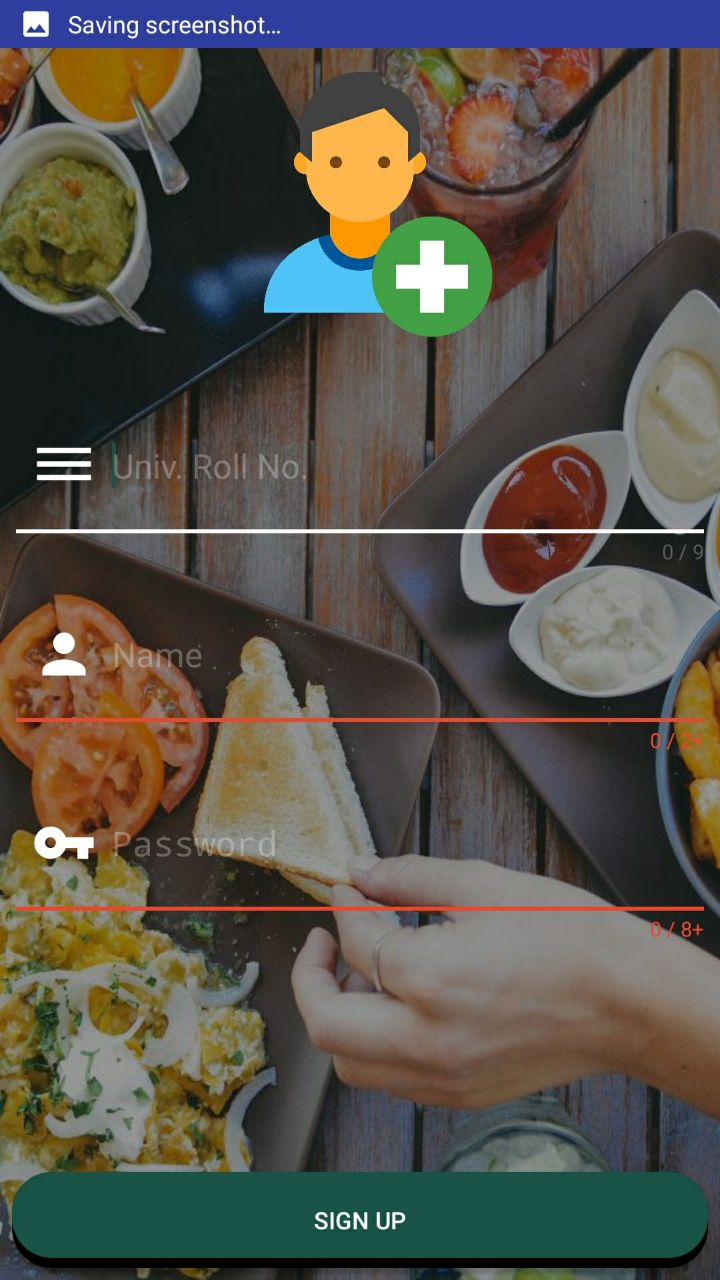


        Fig 10:  Registration page



                  Fig 10 : Menu List

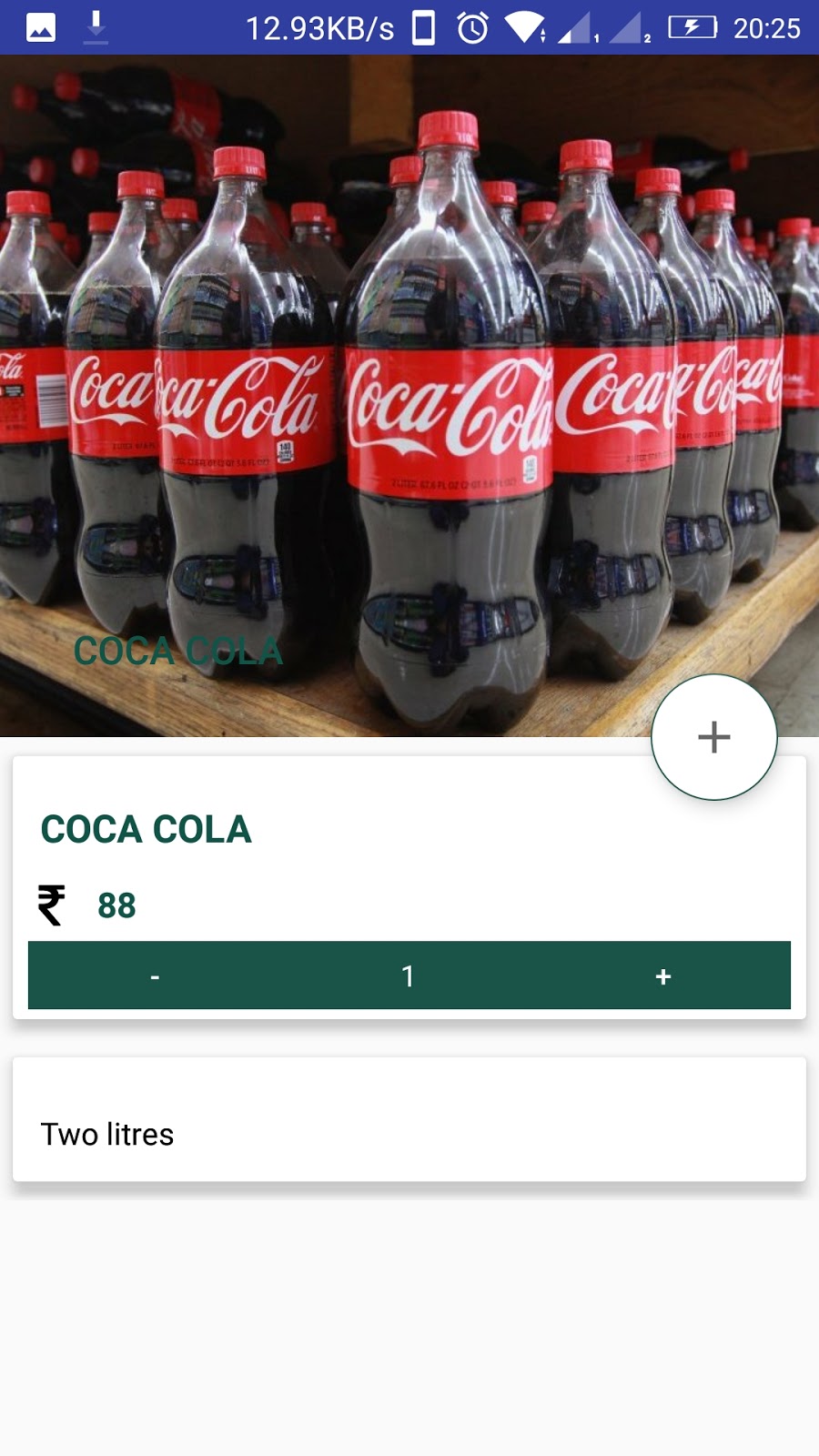


                                                    Fig 10 Add to cart

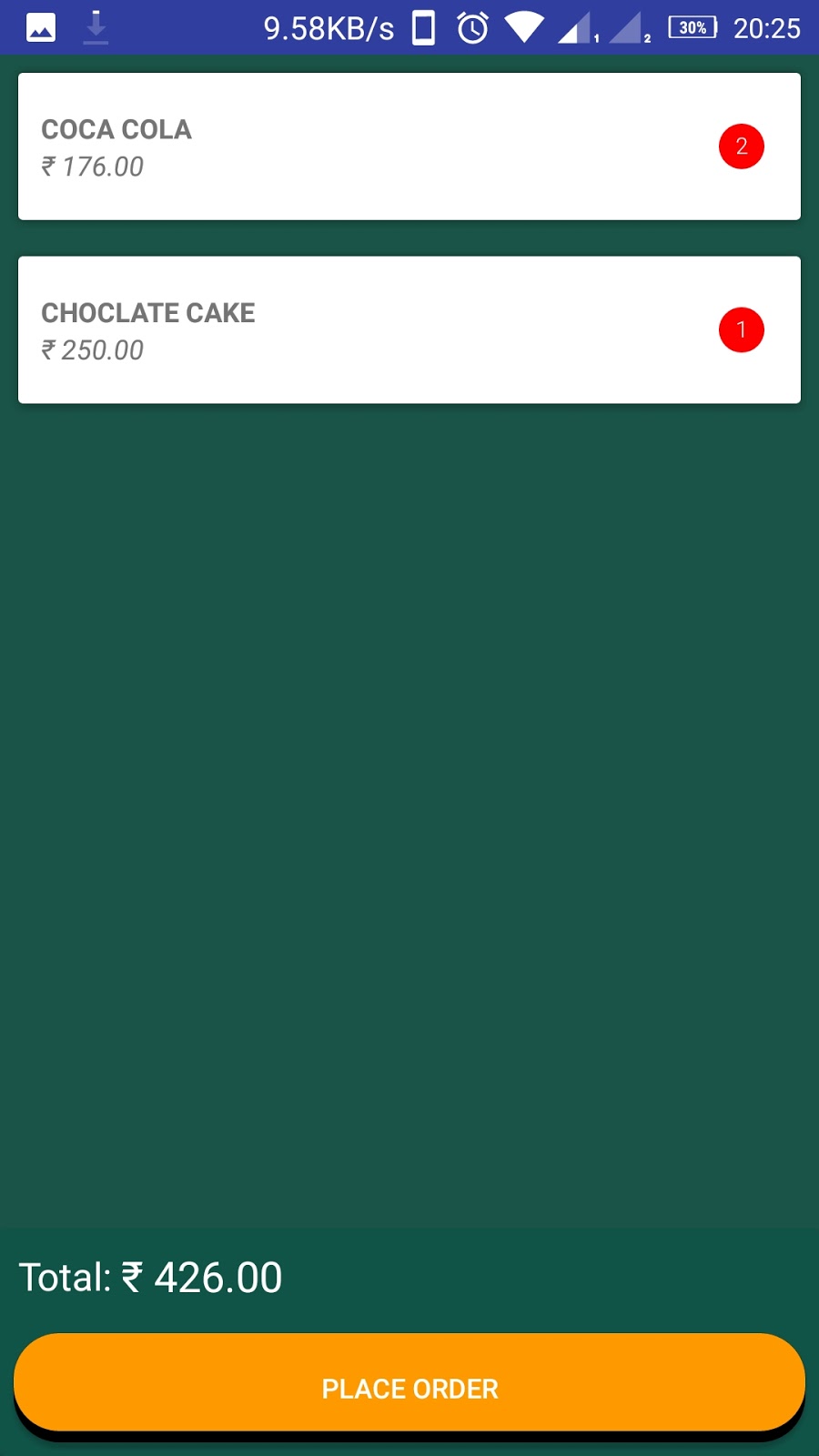


  Fig 10 Order Completion

**BIBLIOGRAPHY**

**WEB LINKS-**

<https://developer.android.com>

<https://www.youtube.com>

<https://www.stackoverflow.com>