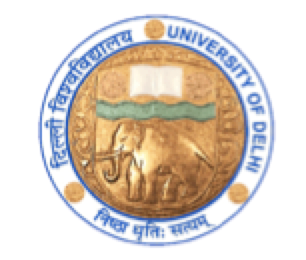
**Web and Android Development**



**Cluster Innovation Centre**

**University of Delhi**

**Ashutosh Jha | Shubham Chauhan**

Nov 2020

**Summer Internship Project**

**Certificate of Originality**

The work embodied in this report entitled **Web and Android Development** has been carried out by **Ashutosh Jha & Shubham Chauhan** for the summer internship project. We declare that the work and language included in this project report is free from any kind of plagiarism.

**(Ashutosh Jha)**  **(Shubham Chauhan)**

**Abstract**

The objective of the summer internship project was to apply theoretical as well as practical knowledge of JAVA, HTML, CSS and JS to build web and android applications. In the internship period we grasped various programming concepts and applied it to make various applications.

1. EsportzGeeks - A web and an android application for gaming related news and updates. The web application is made using HTML, CSS and JS. The android application is made using android framework and firebase
2. EatIt - An android application for food ordering made using android framework and firebase along with braintree payment gateway.
3. NM - An interactive android application for viewing and calculating solutions for different types of numerical methods using a number of factors like no. of iterations, relative error, absolute error etc.

**Table Of Contents**

Abbreviations and terms

1. Introduction
   1. Background of Project
   2. Problem Statement
   3. Goals
2. Analysis of Activity Done
   1. Duration
   2. Project Schedule
      1. Time Schedule
      2. Project Phases
3. System Analysis
   1. System Analysis
   2. Requirement Analysis
      1. Functional requirements
      2. Non-functional requirements
      3. Usability Requirement
      4. Efficiency Requirement
      5. Accuracy
      6. Feasibility study
         1. Economic Feasibility
         2. Technical Feasibility
         3. Operational Feasibility
   3. Technical Requirement
      1. Hardware Requirement
      2. Software Requirement
   4. Input Design
   5. Output Design
4. Implementation
   1. Implementation Tools
   2. Tools Used
      1. HTML
      2. CSS
      3. Javascript
      4. Java
      5. XML
      6. Android framework
      7. Firebase
      8. Braintree (for payment gateway)
   3. System Implementation
   4. System Maintenance
5. Development
   1. Designing Layouts
   2. Implementing the classes and functions
   3. Hosting on Firebase and setting required methods
   4. Integrating Maps SDK and GeoFirestore
   5. Integrating Braintree for payments
6. Testing
   1. Unit Testing
   2. System Testing
   3. Performance Testing
   4. Test Cases
7. Conclusion and Future Enhancement
8. References
9. Bibliography

**Abbreviations and terms**

| Android | An open source platform designed for mobile devices |
| --- | --- |
| OS | Operating system |
| SDK | Software development kit |
| JRE | Java Runtime Environment |
| IDE | Integrated development environment |
| GUI | Graphical user interface |
| Android Emulator | Container for running Android OS along with applications |
| XML | Extensible Markup Language |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| JS | JavaScript |
| RAM | Random Access Memory |
| IT | Information Technology |
| GHz | GigaHertz |
| API | Application Programming Interface |
| ICT | Information and Communication Technology |
| UI | User Interface |
| URL | Universal Resource Locator |
| HD | Hard Disk |
| CPU | Central Processing Unit |

1. **INTRODUCTION**
   1. **Background of Project**

EsportsGeeks is a web and android application for gaming related news and updates. The android application contains an admin and a user app. Each admin is given a username and password, and they can post any article which our readers can read in the user app. The web application is a static set of web pages hosted on Firebase. The content is generated using the claat tool.

EatIt is an android application for food ordering. This android application is also hosted on Firebase and has a no. of features like real-time food tracking, food rating system, payment gateway using Braintree, shippers management system etc.

NM is an interactive android application for viewing and calculating solutions for different types of numerical methods like Newton, Raphson, LU Decomposition, QR Method etc. using a number of factors like no. of iterations, relative error, absolute error etc.

* 1. **Problem Statement**

There is not a single platform for viewing all game related content like gaming updates, esports point tables, esports teams etc. Moreover, as we were learning numerical methods this semester, we didn’t find a single platform where we can solve and view different types of numerical methods for solving single variables, systems of equations etc. So, to overcome these limitations, we have built the proposed solutions.

* 1. **Goals**

Different applications have different goals which are :

* + 1. EsportzGeeks- To develop an web and android application for viewing game related news and updates
    2. EatIt - To develop a food ordering app with various features like food rating system, real-time tracking system, shippers management system etc.
    3. NM - To develop a numerical methods app where the user can view and solve different types of numerical methods.

1. **ANALYSIS OF ACTIVITY DONE**
   1. **Duration**

| **Project** | **EsportzGeeks** | **EatIt** | **NM** |
| --- | --- | --- | --- |
| **Start Date** | June 2020 | July 2020 | September 2020 |
| **End Date** | Present | October 2020 | November 2020 |

* 1. **Project Schedule**
     1. **Time Schedule**

During the internship duration we have done 3 complete projects, and the time spent on various topics is as follow:

| **S.No** | **Task name** | **Duration** |
| --- | --- | --- |
| 1) | Study and Analysis | 15 days |
| 2) | UI Design | 25 days |
| 3) | Database design | 20 days |
| 4) | HTML,CSS | 10 days |
| 5) | JavaScript | 15 days |
| 6) | Responsive Design | 12 days |
| 7) | Bootstrap | 10 days |
| 8) | Logging and Formatting | 7 days |
| 9) | Android | 43 days |
| 10) | Firebase | 12 days |
| 11) | Braintree Payment Gateway | 7 days |

* + 1. **Project Phases**

It is very important to follow up phases as the pre requirements for every next phase is the completion of the previous one. Each project was divided broadly into 4 phases

1. Designing Layouts
2. Implementing the classes and functions
3. Hosting on Firebase and setting required methods
4. Integrating APIs
5. **SYSTEM ANALYSIS**
   1. **System Analysis**

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skillful wall. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

It is the most creative and challenging phase of the system life cycle. The output of this phase is a description of the recommended alternative solution. The steps involved during system analysis process are:

1. Understanding application
2. Planning
3. Scheduling
4. Forming theory background
5. Developing the solution
6. Performing the test analysis
7. Recommending alternative solutions
8. Launching the proposed solution

System analysis can include looking at end-user implementation of a software package or product and involves gathering requirements for the system. In System Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system. The project should address a real world interface design and be implementable. Feasibility Study is a major process in System Analysis. It helps in determining whether the project will yield a desired output with realistic and economic use of available resources.

* 1. **Requirement Analysis**

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

* + 1. **Functional Requirements**

In a development process, functional requirement provides the developer with a blueprint of how the application and its component will behave or function. The functional requirements describe what the application system should do. The functional requirements of the application are as follows:

* + - 1. Esportzgeeks
* The user should be able to view, like, and comment on a news article. No login interface is required.
* The user can filter the news articles according to their interests.
* The admin needs to login and then they can view, edit and create new articles.

So, the following are the functional requirements for Esportzgeeks:

**General**

| G1 | All the data shall be hosted on Firebase for data processing and storage. |
| --- | --- |
| G3 | An app shall provide a User/Admin with all user/admin system functionality (according to access control) |

**User**

| U1 | User shall be able to login or skip from registration to enter the home dashboard |
| --- | --- |
| U2 | Users shall be able to view news articles |
| U3 | Users shall be able to navigate between different articles |
| U4 | Users shall be able to view different categories and subcategories involved |
| U5 | Users shall be able to search for their interests |
| U6 | Users shall be able to like or comment on any article. |
| U7 | Users receive notifications for various news articles. |

**Admin**

| A1 | Admin shall able to Manage users |
| --- | --- |
| A2 | Admin shall able to Manage providers |
| A3 | Admin shall able to CRUD items for Restaurants |
| A4 | Admin shall be able to manage store details |
| A5 | Admin shall able to manage Delivery boy details |
| A6 | Admin shall able to contact between Delivery boy and stores to replace or cancel order if items are not available |

* + - 1. EatIt
* The user can view and order various foods either through food categories, or through restaurants
* Implementation of payment gateway for payments
* Integration of Maps SDK for real-time tracking of orders
* Users can view their previous orders and can cancel the ongoing orders
* The client (restaurant) can view customer’s orders
* The client can modify, add, delete menu items
* Real-time discount system using OR code
* Implementation of Food Rating System
* Implementation of Shippers Management System
* Implementation of Order Management System
* Sending notifications to user and client in various events
* Chat system between user and customer support

So, the following are the identified functional requirements for EatIt :

**General**

| G1 | All the data shall be hosted on Firebase for data processing and storage. |
| --- | --- |
| G2 | A surface app page shall provide a customer with all customer system functionality. |
| G3 | An app shall provide a User/Restaurants with all user/restaurants system functionality (according to access control) |
| G4 | A display shall provide a Delivery Boy with all Delivery boy system functionality. |

**Customer**

| C1 | Customer Shall be able to login or skip from registration to enter the menu dashboard |
| --- | --- |
| C2 | Customer shall be able to view nearby restaurants or menus(Specified Distance) |
| C3 | Customers shall be able to choose their restaurants or food they wish to order stuff |
| C4 | Customer shall be able to view menu and categories and subcategories involved |
| C5 | Customer shall be able to order food and add to cart |
| C6 | Customer shall be able to remove orders from cart |
| C7 | Customer shall be able to navigate between menu and can add items to cart |
| C8 | Customer shall be able to cancel the order |
| C9 | Customer shall able to receive delivery boy details once order picked up |
| C10 | Customer shall able to track the delivery boy details |
| C11 | Customer receives notification for order accepted and once order picked and delivered |

**Restaurants**

| R1 | Stores shall able to CRUD items from menu |
| --- | --- |
| R2 | Stores shall be able to receive orders from customers |
| R3 | Stores shall be able to view the orders which has been ordered by customers |
| R4 | Stores shall be able accept or cancel order depends upon the order received and availability of order |
| R5 | Stores shall able able to receive notifications once order delivered |
| R6 | Stores shall be able to assign delivery boy to deliver order |

**Delivery Boy**

| D1 | Delivery boy shall able to start the shift |
| --- | --- |
| D2 | Delivery boy shall able to receive incoming order request from customers |
| D3 | Delivery boy shall able to acknowledge for the request |
| D4 | Delivery boy shall able to acknowledge for request within a specified amount of time |
| D5 | Delivery boy shall able to reach the restaurant and check with order details |
| D6 | Delivery boy shall able to receive customer location |
| D7 | Delivery boy shall able to pick up and deliver order to customer |

**Admin**

| A1 | Admin shall able to Manage users |
| --- | --- |
| A2 | Admin shall able to Manage providers |
| A3 | Admin shall able to CRUD items for Restaurants |
| A4 | Admin shall be able to manage store details |
| A5 | Admin shall able to manage Delivery boy details |
| A6 | Admin shall able to contact between Delivery boy and stores to replace or cancel order if items are not available |

* + - 1. NM
* Adding Docs feature where user can see various numerical methods
* Adding Numerical Methods Calculator for a single variable, system of equations and interpolation
* Adding a no. of factors to calculate the roots like no of iterations, absolute error, relative error etc

So, the following are the identified functional requirements for application NM :

**General**

| G1 | All the data shall be saved in Internal storage and can be retrieved when a user wants it |
| --- | --- |
| G2 | A surface app page shall provide user with all available operations or methods |
| G3 | The app should provide the functionality to calculate the root(s) according to a specific numerical method. |
| G4 | The app should have criterias for calculating roots like no. of iterations, absolute error, relative error etc. |

* + 1. **Non-functional Requirements**

Non-functional requirements are not concerned with the functions of the system. Instead, they look at the criteria to which the application is expected to conform to. Non-functional requirements can include things like response time and reliability. Some of the Non-functional requirement for the applications are:

1. EsportzGeeks web application should be compatible with the last three major versions of Firefox, Chrome, Safari and Internet Explorer.
2. All the components of the application should be fully loaded within reliable time without downgrading performance.
3. Should be user friendly and content should be readable by all types of users.
4. Should take minimal time, effort, resources or cost to create the web application.
5. Should provide the correct information about all the modules.
6. Should consider the Response times such web page loading, screen open and refresh times of each pages
7. Android applications should strictly follow Material Design guidelines
8. The data should get retrieved from the server even in all types of network connectivity
9. The payment gateway should have a secure endpoint
   * 1. **Usability Requirements**

The web application can be accessed by the users from the Internet using HTML or it is derivative technologies like XML/CSS. The system uses a web browser as an interface and is user friendly. The android applications can be accessed by users on their android devices, given they install the desired apks.

* + 1. **Efficiency Requirement**

Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.

* + 1. **Accuracy**

The system should accurately provide real time information taking into consideration various concurrency issues.The system shall provide 100% access reliability.

* + 1. **Safety Requirements**

The system uses SSL (secured socket layer) in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer’s device containing the user’s password.

* + 1. **Performance Requirement**

The information is refreshed at regular intervals depending upon whether some updates have occurred or not. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

* + 1. **Maintainability and Portability Requirements**

Changes (new parts in addition, password changes, and database changes) must be verified once per day at least. The system should provide automatically notification to patrons.

### **Feasibility Study**

### After studying and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the projects. All projects are feasible – given unlimited resources and infinite time.

### 

### Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

* + 1. **Economic Feasibility**

This is a very important aspect to be considered while developing a project. We decided the technology based on the minimum possible cost factor. All hardware and software cost has to be borne by us.

* + 1. **Technical Feasibility**

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification, and checked if everything was possible using different types of frontend and backend platforms.

* + 1. **Operational Feasibility**

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken are all self-explanatory even to a layman.

* 1. **Technical Requirements**
     1. **Hardware Requirements**
* Minimum 35 MB Hard Disk space for installation of different apks.
* Android Minimum API - 21
* Recommended 2 GB RAM
* Network card
  + 1. **Software Requirements**
* Application system : Webstorm, Android Studio
* Language : HTML, CSS, JavaScript, Java, XML, Android Framework
  1. **Input Design**

Input design is part of overall system design that requires special attention. For designing input data, the data entered should be easy and free from errors. The input forms and dialog boxes in applications are designed using the controls available. Validation is made for each and every data that is entered. Help information is provided for the users.

Input design is the process of converting the user originated inputs to a computer based format. The collection of input data is considered to be the most expensive part of the system design. Since the input has to be planned in such a manner so as to get relevant information, extreme care is taken to obtain pertinent information. The application gives an interface for registering and validating users and the data entered by users are converted to suitable models and uploaded to firebase for further use.

* 1. **Output Design**

Output design of the applications generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. The output is designed in such a way that it is attractive, convenient and informative. As the outputs are the most important sources of information to the users, better design should improve the system’s relationships with us and also will help in decision making. The application presents the information in an interactive, user friendly manner. Moreover, the user can further print Invoices and can interact with the app.

1. **IMPLEMENTATION**
   1. **Implementation Tools**

The various implementation tools are listed below:

* Operating System - Windows 10
* Development environment - HTML5, CSS3, Bootstrap, JavaScript, Java, XML, Android SDK
* IDE - Webstorm, Android Studio 4.1.1
  1. **Tools Used**

The front is an abstraction, simplifying the underlying component by providing a [user-friendly](https://en.wikipedia.org/wiki/User-friendly) interface. There are several tools available that can be used to develop the front end of a web and android application

* + 1. **HTML**

HTML (Hypertext Markup Language) is the most basic building block of the Web. It describes and defines the content of a webpage. "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web. HTML uses "markup" to annotate text, images, and other content for display in a Web Browser.

* + 1. **CSS**

Cascading Style Sheets (CSS) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language). CSS is designed primarily to enable [the separation of document content from document presentation](https://en.wikipedia.org/wiki/Separation_of_presentation_and_content), including aspects such as the [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

* + 1. **Javascript**

JavaScript is the client side scripting language of the web. It's one of the most popular and in demand skills in today's job market for good reason. JavaScript not only enables you to add powerful interactions to websites, but is also the foundation of a lot of commonly used libraries like jQuery.

* + 1. **Android framework**

Android is one of the Open source platforms. It is created by Google and owned by Open Handset Alliance. It is designed with the goal “accelerate innovation in mobile”. As such android has taken over a field of mobile innovation. It is definitely a free and open platform that differs hardware from software that runs on it. It results in much more devices running the same application. Also it gives the possibility of a friendlier environment for developers and consumers. Android is a complete software package for a mobile device. Since the beginning the android team offers the developing kit (tools and frameworks) for creating mobile applications quick and easily as possible. In some cases you do not especially need an android phone but you are very welcome to have one. It can work right out of the box, but of course users can customize it for their particular needs. For manufacturers it is a ready and free solution for their devices. Except specific drivers, the android community provides everything else to create their devices.

* + 1. **Firebase**

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of November 2020, the Firebase platform has 18 products, which are used by 1.5 million apps. We used it for hosting purposes. The features we used in our app are

1. Authentication Methods
2. Firestore
3. Realtime Database
4. Cloud Storage
5. Cloud Functions
6. Cloud Messaging

* + 1. **Braintree (for payment gateway)**

Braintree is a full-stack payments platform that makes it easy to accept payments in an app or website. Our service replaces the traditional model of sourcing a payment gateway and merchant account from different providers.

* 1. **System Implementation**

System implementation is the important stage of a project when the theoretical design is tuned into practical system. The main stages in the system implementation are as follows:

1. Planning
2. Training
3. System testing
4. Changeover planning

EsportzGeeks will implement article reading, liking, and commenting features. The admin can login and can view, edit and write articles.

EatIt android application lets users register themselves and order foods according to various categories or menus. The order is sent to the restaurant via client app. The delivery boy is notified using Shippers app and the customer can cancel, modify or track their orders in real time. Users can also rate different foods. The payment gateway is implemented using Braintree payments using a secure endpoint.

NM application lets users view different types of numerical methods using the DOCS grid and lets users solve various problems using numerical methods according to some defined criterias.

* 1. **System Maintenance**

Software maintenance is far more than finding mistakes. Provision must be made for environment changes, which may affect either the computer, or other parts of the computer based systems. Such activity is normally called maintenance. It includes both the improvement of the system functions and the corrections of faults, which arise during the operation of a new system. It may involve the continuing involvement of a large proportion of computer department resources. The main task may be to adapt existing systems in a changing environment. Backup for the entire database files are taken and stored in cloud systems so that it is possible to restore the system at the earliest. If there is a breakdown or collapse, then the system gives provision to restore database files. Storing data in a separate secondary device leads to an effective and efficient maintenance of the system.

1. **Development**
   1. **Designing Layouts**

Layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and ViewGroup objects.

The View objects are usually called "widgets" and can be one of many subclasses, such as Button or TextView. The ViewGroup objects are usually called "layouts" and can be one of many types that provide a different layout structure, such as LinearLayout or ConstraintLayout.

You can declare a layout in two ways:

* **Declare UI elements in XML**: Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.
* **Instantiate layout elements at runtime** : App can create View and ViewGroup objects (and manipulate their properties) programmatically.

Declaring your UI in XML allows you to separate the presentation of your app from the code that controls its behavior. Using XML files also makes it easy to provide different layouts for different screen sizes and orientations. So, the main elements used in our project were:-

1. Parent Views- Linear View, Relative View and Constraint view for holding widgets and child elements
2. Child Elements

* Buttons:- Whenever user clicks it, then it performs a specific action
* EditText:- To take user Input
* CardView:- Show information inside cards that have a consistent look across the platform
* GridLayout:- a Layout manager that lays out a container's components in a rectangular grid. The container is divided into equal-sized rectangles, and one component is placed in each rectangle. Used along with a card view.
* TextView
* ImageView
* PinView:- Widget for entering PIN

1. Sub-child elements- Combination of the above child elements
   1. **Implementing the classes and functions**

We used various classes and methods in our project which is beyond the scope of this report. The main methods and classes used were:-

1. setOnClickListener- It helps us to link a listener with certain attributes. setOnClickListener is a method in Android basically used with buttons, image buttons etc.
2. addOnSucessListeners
3. addOnFailureListener
4. Toast - For displaying user a message
5. Intent- For taking user to other activities
6. verifyCode method for verification of OTP
7. onMapReady for displaying map
8. onLocationChanged
9. DocumentReference- for referring to the firestore database according to the document ID etc.
   1. **Hosting on Firebase and setting required methods**

Firebase gives us functionality like analytics, databases, messaging and crash reporting so we can move quickly and focus on the users. Firebase is built on Google infrastructure and scales automatically, for even the largest apps. The steps involved were:

* We need to set up a Firebase Account and create a new project.
* Then provide the necessary details and provide the apps SHA-1 fingerprint.
* Add the firebase configuration files and plugins.
* Add firebase SDK to the app
* In the console under sign-in method, enable Phone authentication and implement the required methods according to the Firebase Docs.
* Then, in the database section, select Firestore and the region we wish to store our data. Then make a collection and accordingly document. Each document has a specific ID. In our case, it is the user’s phone no. We use DocumentReference and CollectionReference class to refer to documents (We have to give document Id) and collection(Provide Collections name) respectively.
* We can then use various methods to store and retrieve data
  1. **Integrating Maps SDK and GeoFirestore**

With the Maps SDK for Android, we can add maps based on Google Maps data to our application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. To integrate it one needs an API key which they can get on Google Cloud Platform Console. Enable the APIs you wish to integrate. We have already supplied the enabled APIs in our application. A GeoFirestore object is used to read and write geo location data to the Firestore database and to create queries. So, in simpler terms we used GeoFirestore to store user location in our database according to a specific document ID.

* 1. **Integrating Braintree for payments**

There are a number of ways to collect customer payment information via the client SDK. Firstly, set up the Android SDK. Get a client access token and start accepting payments. Moreover, drop-in UI is the easiest way to get a full-featured checkout with credit cards and paypal payments.

1. **Testing**

Testing is a method of assessing the functionality of a [program](http://searchsoftwarequality.techtarget.com/definition/program). Testing is a set of processes aimed at investigating, evaluating and ascertaining the completeness and quality of a project. Testing refers to the process of implementing all or part of the system with the intent of finding errors. It is performed in order to find the bugs or errors in the system and minimize it. In general, testing is finding out how well something works .Testing is more than just debugging.

* 1. **Unit Testing**

Each division class of every page or layout is tested in the browser or android device, whichever applicable. Inspecting HTML and XML, and modifying style and layout in real-time.

* 1. **System Testing**

After completing the overall application design and development it is tested for error by uploading the file in w3c, which shows errors along with the warnings. It also provides the validation output errors with detailed descriptions along with the code as well as line and column number in which the error occurred. W3c provides the validation by URI, by file upload and by direct input.

* 1. **Performance Testing**

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

* 1. **Test Cases**

The various test cases for EatIt are :

| Test Case ID. | 1 |
| --- | --- |
| Test Case Name | Registration Account |
| Test Case Description | This test case tests the registration process |
| Dependency or Prerequisite | Connection to the Database |
| Steps | 1. Application displays the link to registration on start 2. User press the registration link tab 3. User is asked to fill a form consisting of details like Name, Email, Mobile No., & Address. 4. If the information matches the required format the user can login into his account. |
| Expected Result | Successful registration of user Information |
| Actual Result | 1. Successful registration on providing the information 2. Registration fails if incorrect information is provided |
| Estimated Time | Depends upon the user (approx. 2 min) |

| Test Case ID. | 2 |
| --- | --- |
| Test Case Name | Login Account |
| Test Case Description | This test case tests the login process |
| Dependency or Prerequisite | 1. Connection to database is properly established 2. Account is created or not. |
| Steps | 1. Application displays the link to login on start 2. User press the link 3. New form asking phone number and otp appears 4. If correct information is entered user can login 5. After logging in, the user can look at his profile & will have access to place order. |
| Expected Result | Successful login if phone no and OTP is correct |
| Actual Result | 1. Logged in on correct information. 2. Login fails on incorrect information. |
| Estimated Time | 1-2 secs depending upon the internet speed. |

| Test Case ID. | 3 |
| --- | --- |
| Test Case Name | Searching Product |
| Test Case Description | This test case tests searching process of a product |
| Dependency or Prerequisite | Product Database Record |
| Steps | 1. User needs to click on the search tab 2. Type the desired product he/she wants to search 3. The system will show the products which the user has searched for. |
| Expected Result | Successful results of the product searched |
| Actual Result | 1. Correct or similar products displayed. 2. No products displayed. |
| Estimated Time | Less than 5 sec depending on the network speed |

| Test Case ID. | 4 |
| --- | --- |
| Test Case Name | View Restaurants and Menus |
| Test Case Description | This test case tests the available stores. |
| Dependency or Prerequisite | Connected to Database |
| Steps | 1. Application displays a search tab to explore stores within. 2. Users can check in to various stores according to the needs. 3. The system will show the stores among which the product is available. |
| Expected Result | Successful results of the Store searched |
| Actual Result | 1. Correct or similar Store displayed. 2. No Store displayed. |
| Estimated Time | Less than 5 sec. |

| Test Case ID. | 5 |
| --- | --- |
| Test Case Name | Order Placement |
| Test Case Description | This test case tests the Order Placement process |
| Dependency or Prerequisite | User cart Data |
| Steps | 1. User needs to select the product which they want to order. 2. User needs to add his address on which they want the product to be delivered. 3. Users can set the payment method. 4. By clicking on the place order tab the order will be placed. |
| Expected Result | Order to be successfully placed. |
| Actual Result | 1. Order will be successfully placed if the address is in the area we deliver the products and the information is provided correctly. 2. Order will not be placed if the address is not in the region we deliver the products or any information is incorrect. |
| Estimated Time | Less than minute |

| Test Case ID. | 6 |
| --- | --- |
| Test Case Name | Order Tracking |
| Test Case Description | This test case tests the Orders Movement from store to User. |
| Dependency or Prerequisite | Data of the Order Placed |
| Steps | 1. User can track the location of their products 2. They can even call the delivery boy allotted for their respective products 3. The product will be delivered at your doorstep. |
| Expected Result | Order to be tracked accurately and delivered. |
| Actual Result | 1. Order to be tracked accurately. 2. Order tracking may vary depending upon the connectivity with the person delivering the order. |
| Estimated Time | Order Tracking : less than 1 min  Order Delivering: within hours |

1. **CONCLUSION**

In this Project, prototypes of the applications were developed. The application can connect to the server and retrieve information from the server, wherever applicable. It can be installed and functions on several devices at the same time. The prototype is working on an android platform and made on the base of Java framework. It uses Firebase to store and to receive information. The information is stored on Firebase and can be accessed any time.

During this project we have gone through several phases: creating project blueprint, forming theory background, creating an environment for developing application, designing model of application, implementing prototype, performing testing. Methods which were used during the process are literature review and rapid application development. On the basis of requirement analysis we have created layouts for prototype and were successfully able to deploy our app

1. **SCOPE FOR FUTURE DEVELOPMENT**
   1. **EsportzGeeks**

The web and android application can be enhanced to view, add and modify points tables and participating teams for various esports competitions. Moreover, various new game trailers sections can be added to the android application. Easy navigation view can be implemented for easy switch between various components of the app

* 1. **EatIt**

The eatit app has many future scope enhancements like automatic registration of users, integration of Places API, integration of a chat bot for various common queries, integration of twitter and facebook login system, integration of sending payments, ledger or SOL documents.

* 1. **NM**

Graph features can be implemented in the application. The various solution steps can be shown to users.

1. **REFERENCES**
   1. Udacity Android Developer Program (<https://udacity.com>)
   2. Android Developers ([https://developer.android.com](http://developer.android.com))
   3. Dawidowicz, Paula 2010 Literature Reviews Made Easy: A Quick Guide to Success. IAP.
   4. Gargenta, Marko 2011 Learning Android. O’Reilly Media, Inc.
   5. StackOverflow (<https://stackoverflow.com>)
   6. Material Design for Android (<https://material.io>)
   7. Firebase Docs (<https://firebase.google.com/docs>)
   8. Maps SDK for Android (<https://cloud.google.com/maps-platform>)
   9. Github ( <https://github.com> )