**Software Requirements Specification**

**For**

**KnackBag**

**Version 1.0 approved**

**Prepared by:**

**Akshay Ratnaker (0801IT161010)**

**Hritik Jain (0801IT161029)**

**Lavi Junwal (0801IT161034)**

**Naman Juneja (0801IT161039)**

**Table of Contents**

**Table of Contents** [**ii**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.gjdgxs)

**Revision History** [**ii**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.30j0zll)

**1. Introduction** [**1**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.1fob9te)

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope          1

1.5 References

**2. Overall Description** [**2**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.1t3h5sf)

2.1 Product Perspective [2](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.4d34og8)

2.2 Product Functions [2](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.2s8eyo1)

2.3 User Classes and Characteristics [2](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.17dp8vu)

2.4 Operating Environment [2](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.3rdcrjn)

2.5 Design and Implementation Constraints [2](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.26in1rg)

2.6 User Documentation [2](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.lnxbz9)

2.7 Assumptions and Dependencies [3](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.35nkun2)

**3. External Interface Requirements** [**3**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.1ksv4uv)

3.1 User Interfaces [3](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.44sinio)

3.2 Hardware Interfaces [3](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.2jxsxqh)

3.3 Software Interfaces [3](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.z337ya)

3.4 Communications Interfaces [3](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.3j2qqm3)

**4. System Features** [**4**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.1y810tw)

4.1 System Feature 1 [4](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.4i7ojhp)

4.2 System Feature 2 (and so on) [4](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.2xcytpi)

**5. Other Non-functional Requirements** [**4**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.1ci93xb)

5.1 Performance Requirements [4](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.3whwml4)

5.2 Safety Requirements [5](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.2bn6wsx)

5.3 Security Requirements [5](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.qsh70q)

5.4 Software Quality Attributes [5](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.3as4poj)

5.5 Business Rules [5](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.1pxezwc)

**6. Other Requirements** [**5**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.49x2ik5)

**Appendix B: Analysis Models** [**5**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.147n2zr)

**Appendix C: To Be Determined List** [**6**](https://docs.google.com/document/d/1hhcFLvWNWZ0sW-K68ul6gcP0yNkvGScS8MpNfegbH1A/edit#heading=h.3o7alnk) ****

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

* 1. **Purpose**

KnackBag is basically a web-based chat-bot that enables the user to enter the various specifications of a laptop/mobile phone. Based on these entries, the best match for the following specifications is generated. This saves a lot of time as now people won’t have to surf the net for hours and visit different websites for selecting a laptop/mobile phone of the required specifications.

* 1. **Intended Audience**

KnackBag is mainly for people who know the specifications of the device they require as well as their budget but don’t know which model will serve their purpose in the best possible manner. It would also attract gadget-freaks who also want to know what’s new and always like to keep a check on the technology.

**1.3 Product Scope**

KnackBag focuses on providing the best matches for devices (laptop/mobile phones initially) based on the specifications and the budget entered by the user. It is basically an interactive chat-bot that firstly allows the user to choose the device he is looking for(laptop/mobile phones initially).Then it asks the users to provide specifications(storage, display, ram, processor etc.),brand of the device and the budget of the user and in turn, it provides the user with the model that fulfils all the conditions that are put up by the user in the best possible way. In near future, it would also give best buy links for the best match. It can even contain videos that compare two specific models that are somewhat equivalent and fulfil all the conditions put up by the user. Our aim is to make KnackBag globally recognized and make it barometer of the technological market.

* 1. **References**

1. Chatter.io: https://www.chatteron.io/

**2. Overall Description**

**2.1 Product Perspective**

The system will be Web Based in which we will have a website consisting of details of the product, how it will assist user in finding best possible product i.e. mobile phone or laptop on the basis of specifications required by the user. Website will consist of a chatbot. User can interact with chatbot in order to find information about the products they wish to buy. User can ask for any help and query by requesting a scheduled call. Website will be main interface on which user can interact with chatbot, whole functionality of the project will be accessed using website.

* 1. **Product Functions**

Functionalities offered by the Product:

* Login/Register user.
* Store user details into database.
* Store product details into database.
* Choose from products offered by system to enquire.
* Chat with Chatbot.
* Input various specifications for the product.
* Ask queries with system about product selected.
* Get queries from user and process it to get output.
* Produce result for user by fetching data from database.
* Schedule call for any other query or help.
  1. **User Classes and Characteristics**

Users of the system should be able to get information about the most relevant product they wish to buy based on the series of inputs given by them for the specifications they want in a product. Users must be able to ask queries about the product description. User must be a registered user in order to access system. Interface provides user to register and login to interact with the system.

Different User Classes are:

The customer should be able to do the following functions:  
  
    Register/login

   •Username  
    • Email-Id  
    • Password  
  
    Chat with Chatbot   
    • Text conversion.  
    • Selecting options provided by chatbot.

   Ask query about product

   • Select product: Mobile or Laptop.  
    • Enquire about product.

   Schedule Call

   • Register a call to enquire more about product or any other query.

The Chatbot should have following management functionalities:  
  
    CHATBOT FUNCTIONS.  
    • Give choices to users to enquire about product.  
    • Ask for product specifications from user.  
    • Get user query.  
    • Process query and produce result from database.  
    • Give result to user.

   ADMINISTRATIVE  
    • Add/Delete user information  
    • Add/Delete/Update product details  
    • Confirm call schedule  
    

**4. Operating Environment**

This a web based application so it will support all the systems which has a web browsers and the systems must have internet connectivity with them.

**5. Design and Implementation Constraints**

The issue which we faced while implementing is that the bot will not work if user gives some input. He/she will be required to select from the options provided to him. Also they will be required to have a Facebook account as it is a messenger bot. The technologies which we have used is Html5, CSS3,bootstrap,php,Mysql, The user can simultaneously chat with our bot and schedule a call; There is one to one communication between bot and the user. No other can see the chat between them. Also, the login credentials provided by the user will be secured and will not be shared with anyone; security considerations; We have used extreme programming for our software.

**6. User Documentation**

We have provide a guide for our website in the form of a pdf. It will contain how to use the bot and how to schedule a call.

* 1. **Assumptions and Dependencies**

Assumption:

* User has basic knowledge of internet.
* User should knowledge English
* User should have Facebook account.

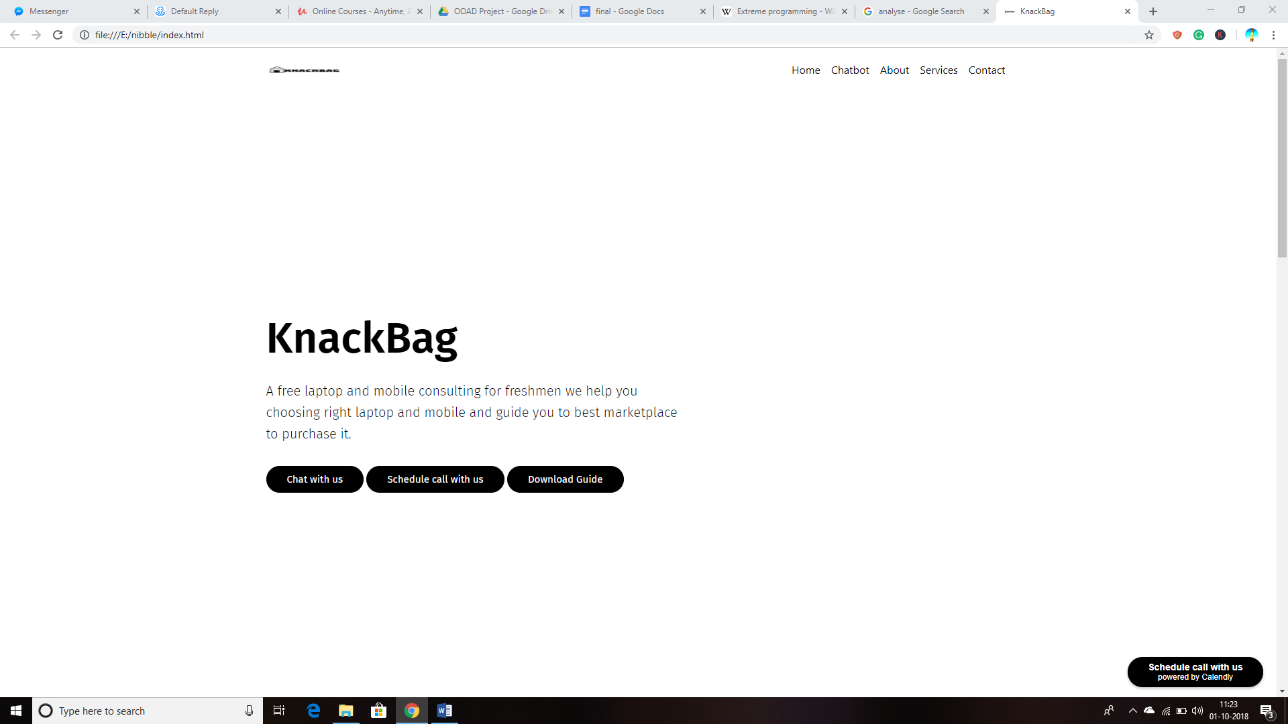
**Dependencies**:

* Internet connection
* User should have valid E-mail Id and Phone number.

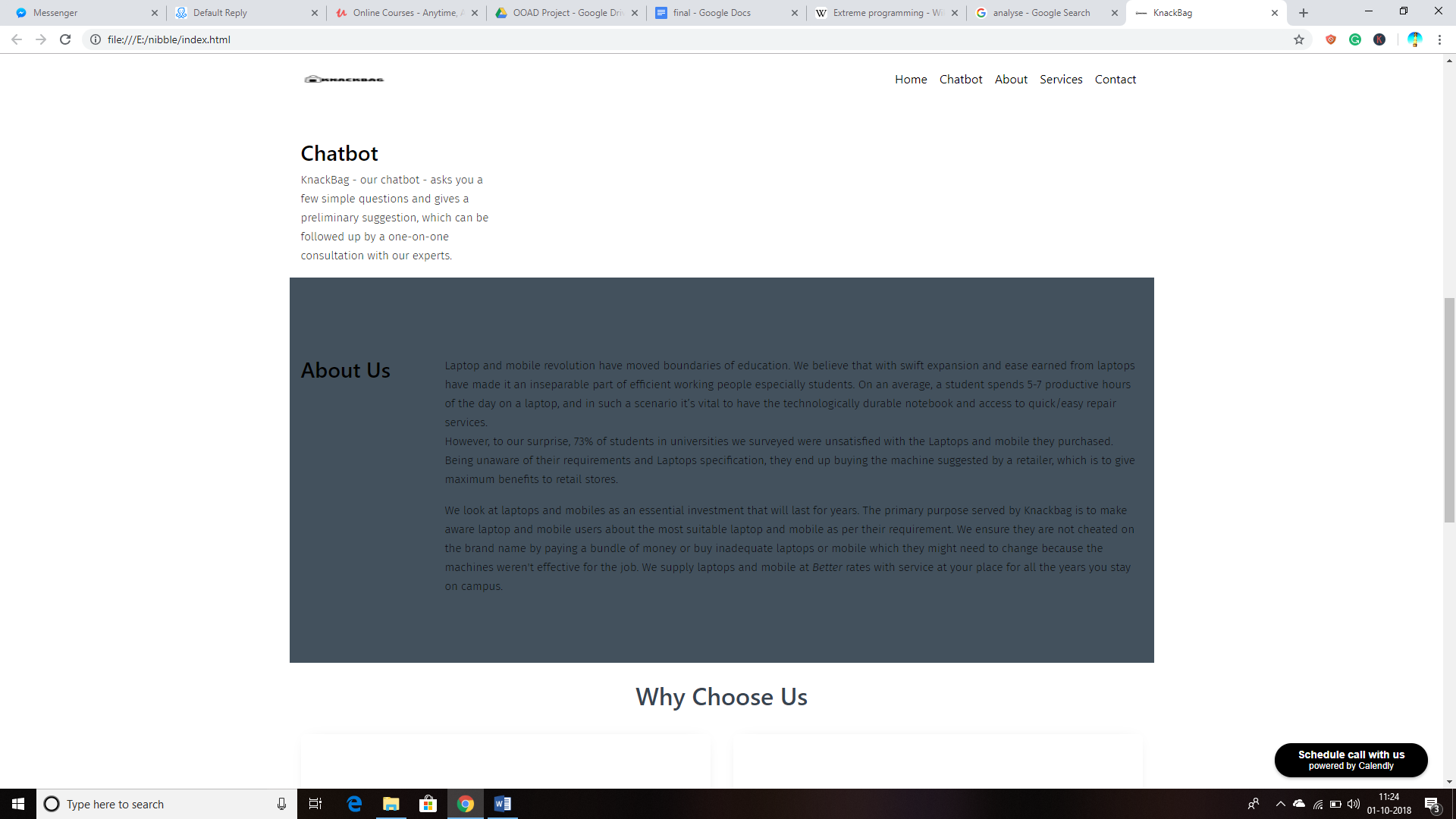
**3**. **External Interface Requirements**

* 1. **User Interfaces**

We have described the logical characteristics of each interface between the software product and the users. This includes sample screen images, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on.

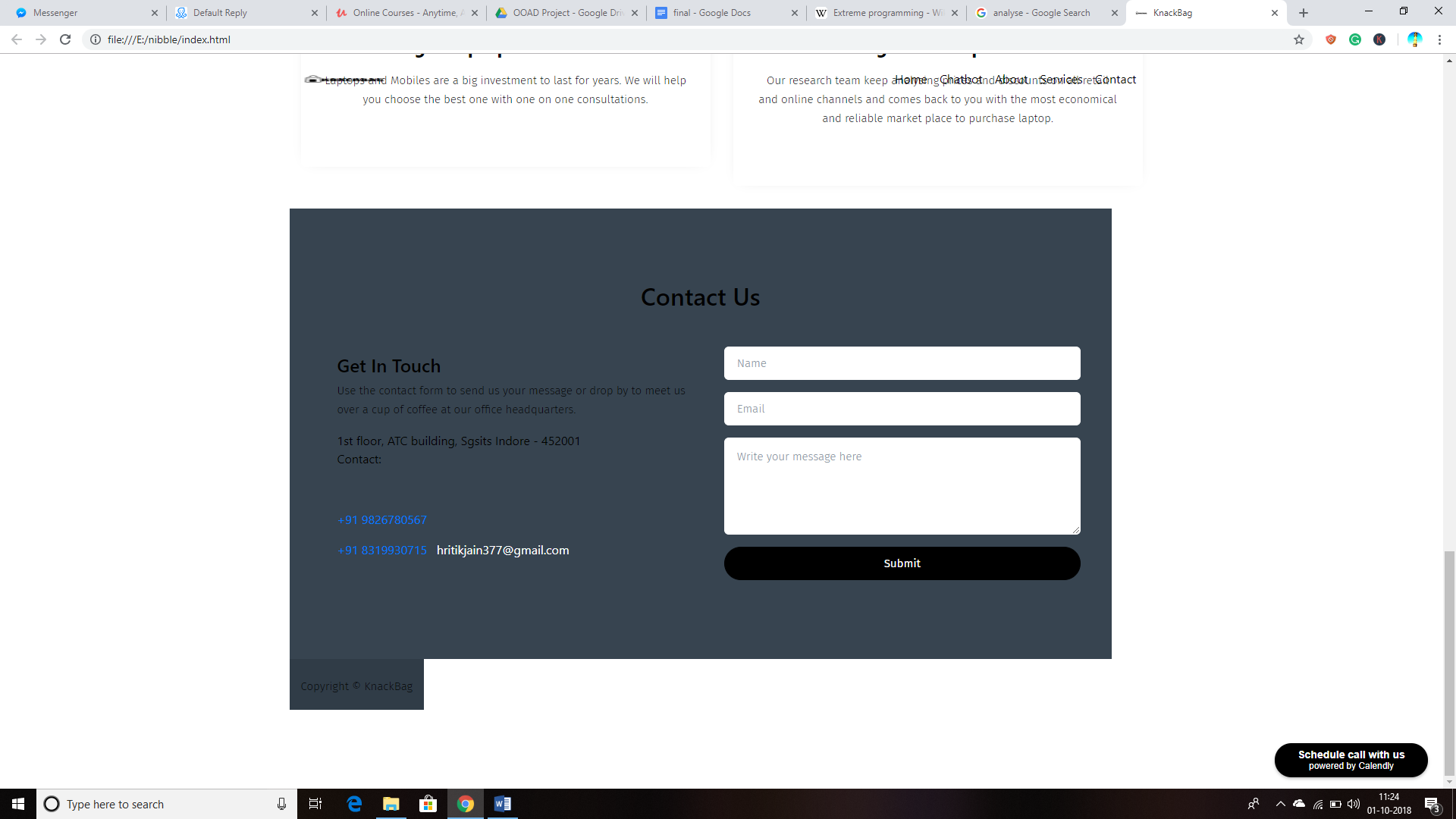


Description: The front page will look like this. It will include Buttons for Chat with Bot, Schedule a call, and download guide.

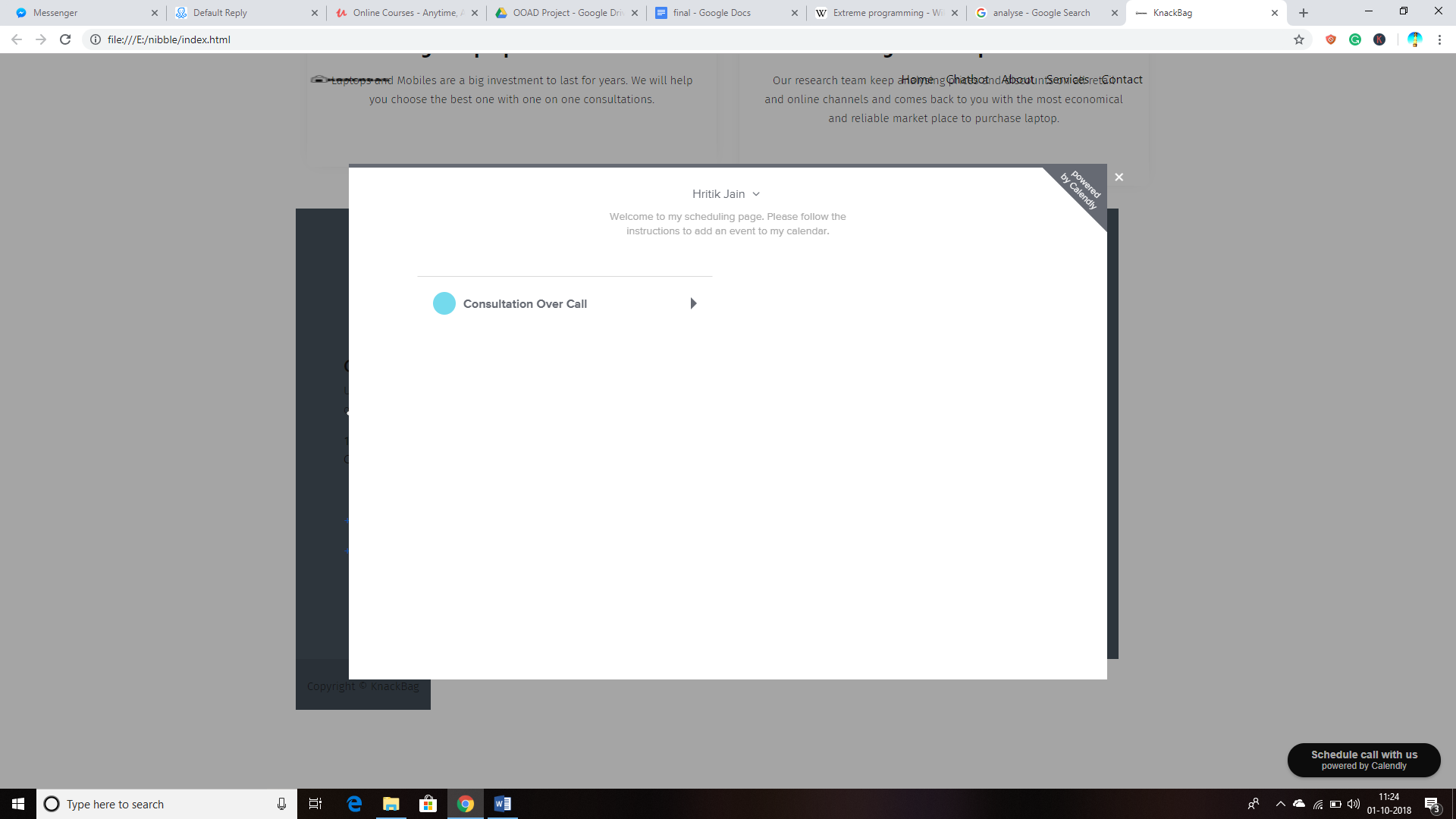


Need of chatbot:

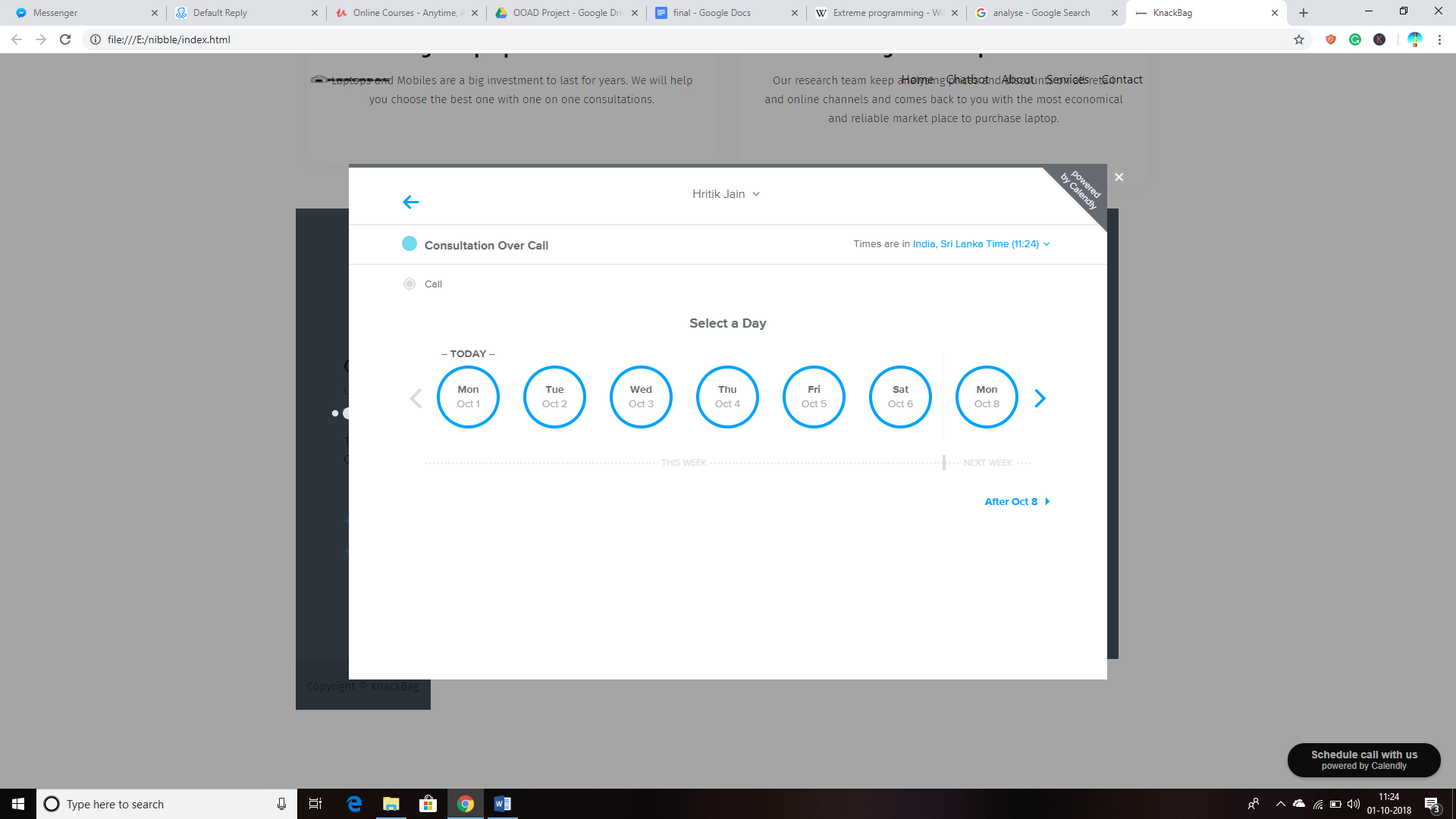
According to stats it is found that 70% of people are not satisfied with the laptops/mobiles they own as they get confused with specifications they wish to get according to need. This chatbot will help users to select appropriate mobile/laptop according to specifications entered by them.



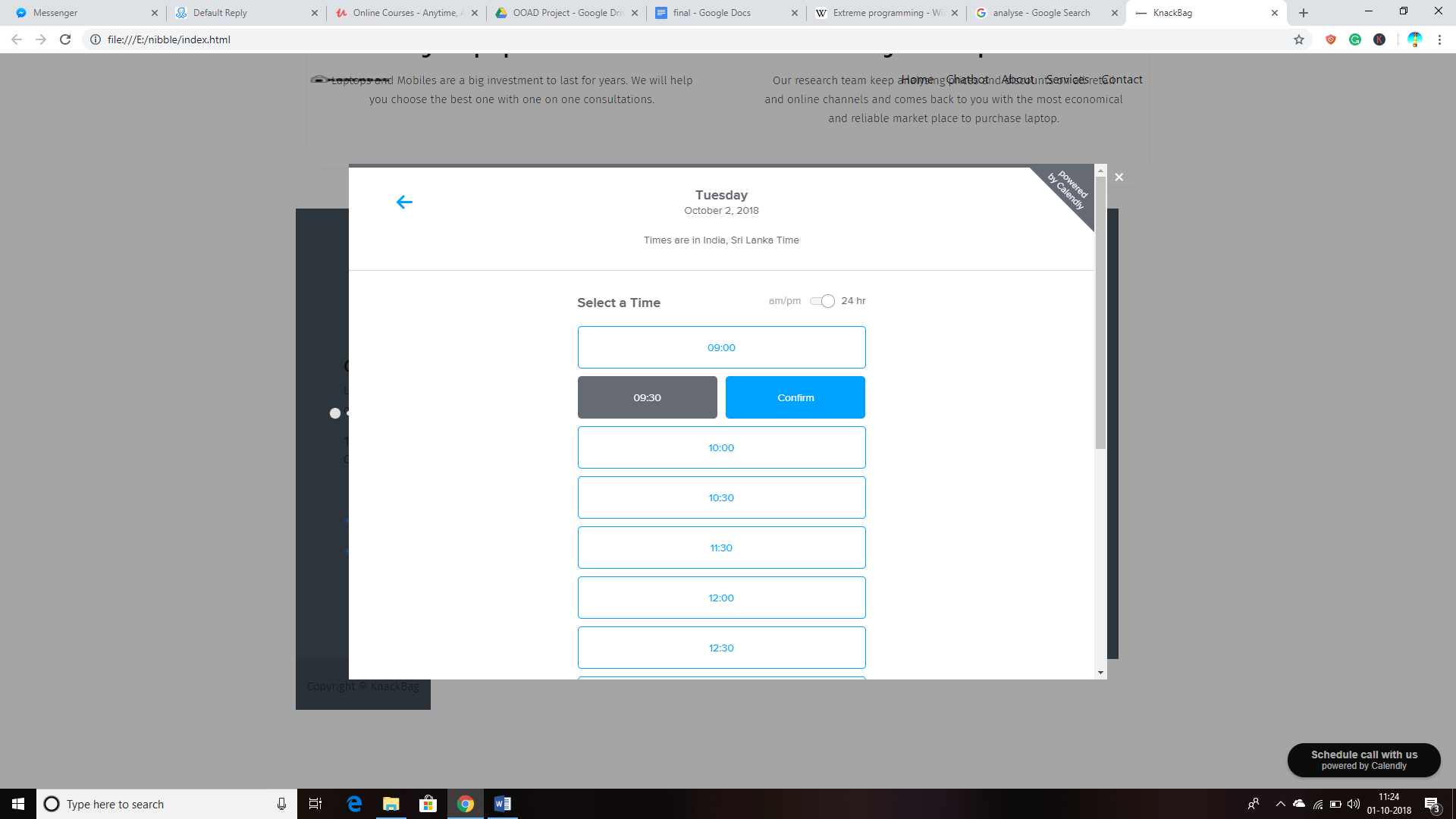
Description: This scenario will help user to provide feedback.



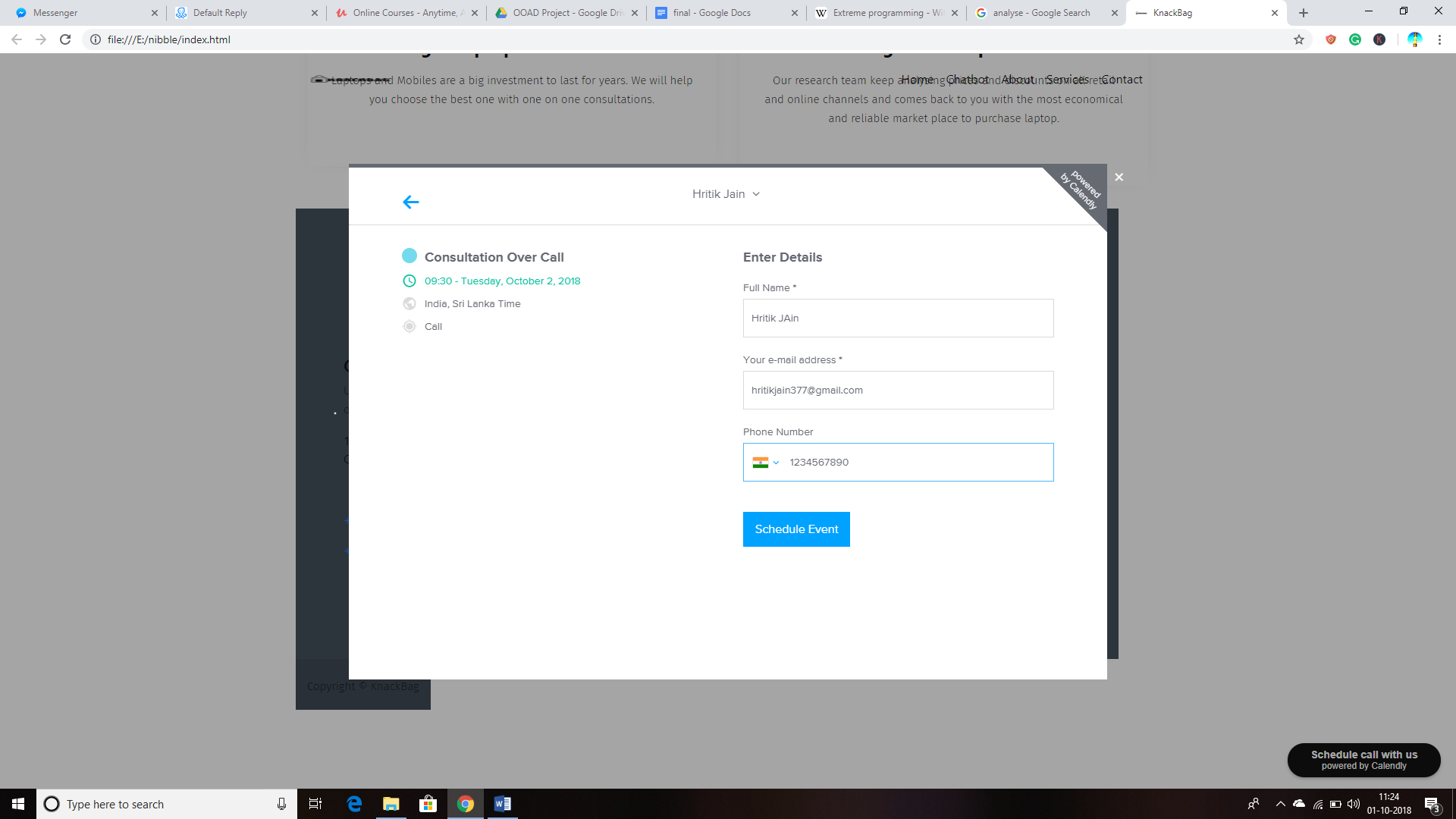
Description: User can schedule a call in case of any query and problem.



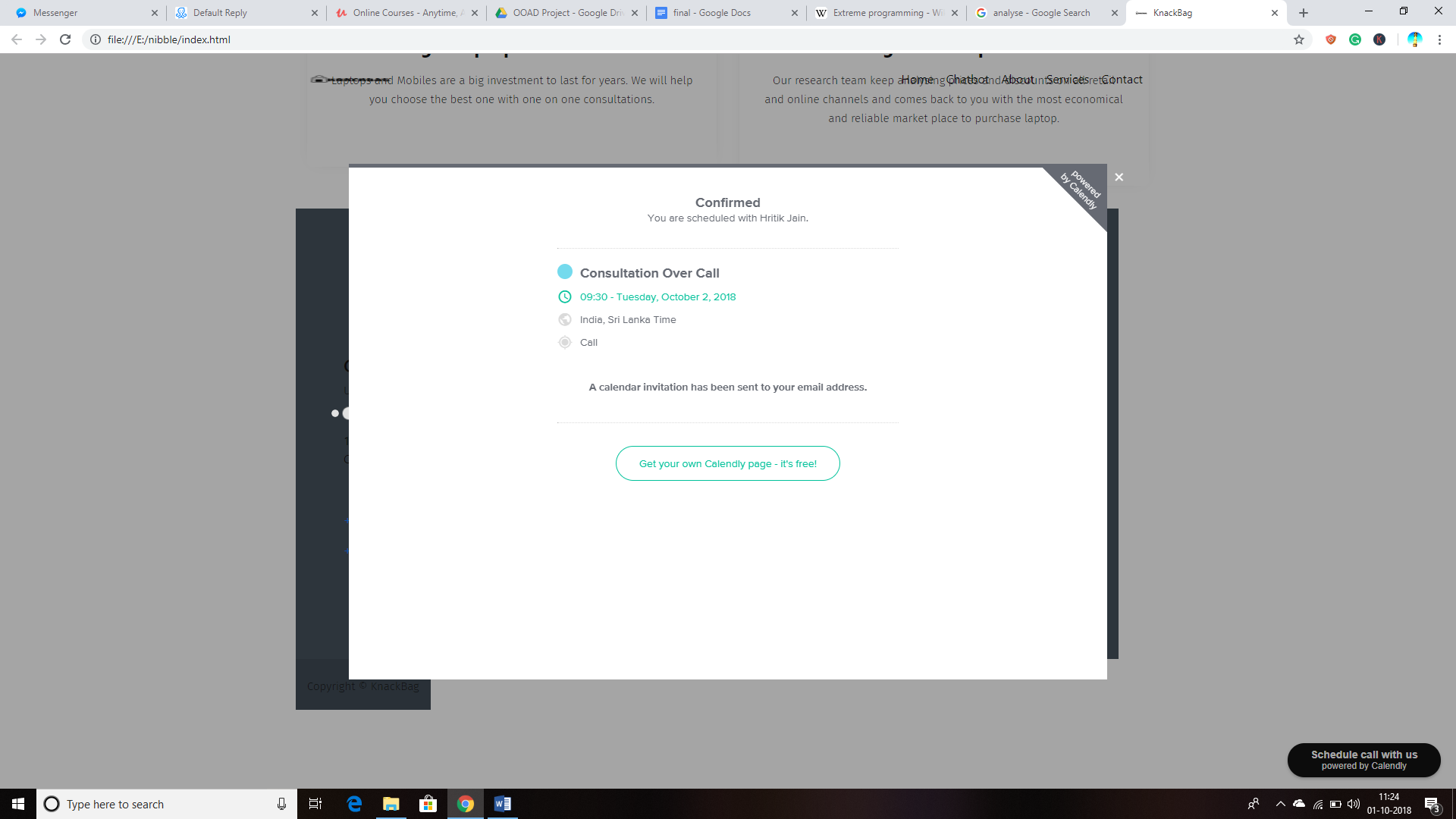
Description: User will select appropriate date to schedule a call.



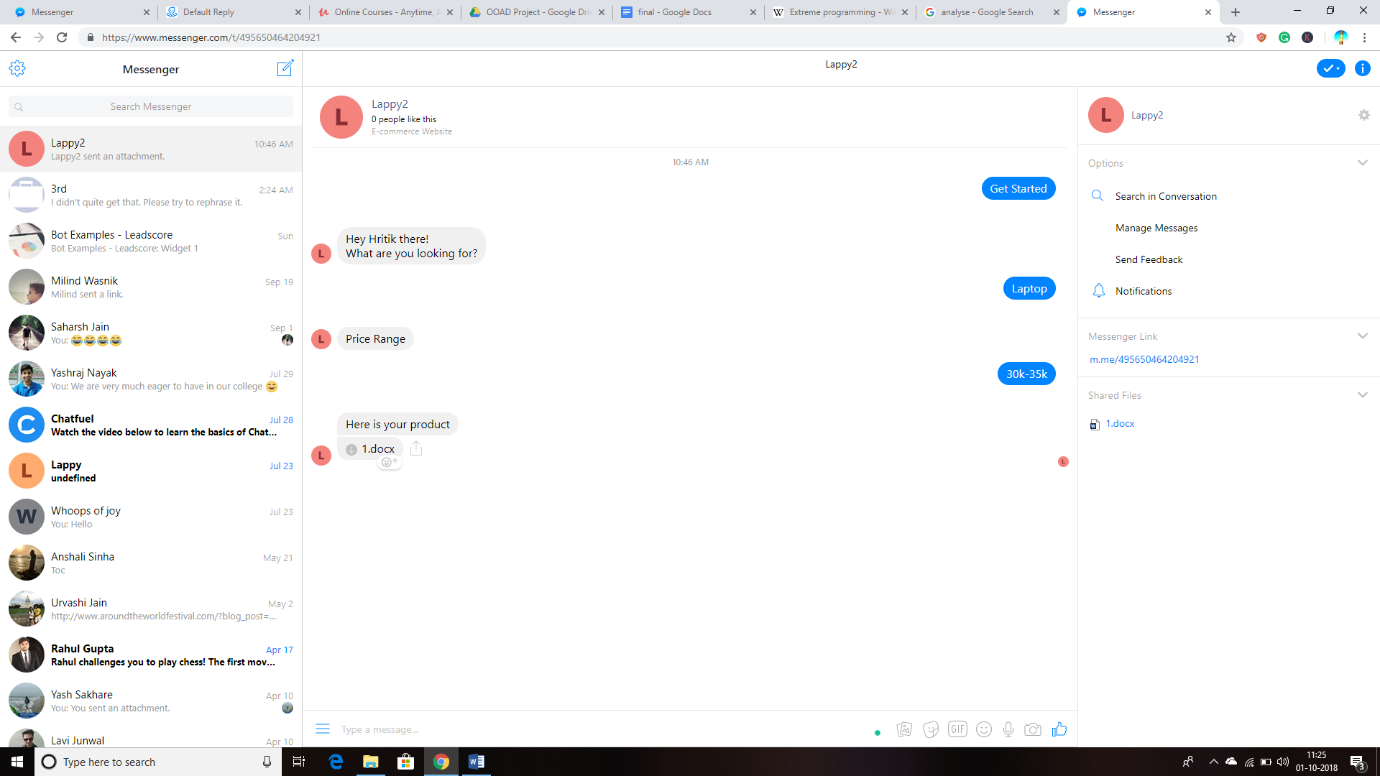
Description: User will select appropriate time.



Description: This scenario will prompt user to enter Name, Email and Phone no. and click on schedule call button.



Description: An confirmation mail will be sent to user, describing all necessary details.



Description: Working of ChatBot Application.

**2. Hardware Interfaces**

This website can be accessed by both mobile and laptop.

* Windows/ Andriod/ iOS/ Linux
* A browser which supports Javascript, HTML.

**3. Software Interfaces**

|  |  |
| --- | --- |
| Software Used | Description |
| Operating System | We have chosen Windows OS for its best support and user friendliness |
| Database | MongoDB – We have stored mobile and laptop specification in MongoDB database with Mobile and laptop collection. |
| Node.js  HTML  CSS  Bootstrap | To implement the project we have chosen node.js environment and express.js module to build API  For front end we chose HTML CSS and bootstrap. |

**Communications Interfaces**

For communication between user and customer, E-Mail address and phone number will be required. We have facebook messenger for communication between user and bot.

**System Features**

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by **use case**, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

* 1. **System Feature 1**

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

* 1. **System Feature 2 (and so on)**

1. **Other Non-functional Requirements**

**1. Performance Requirements**

1. Response time: Systems response time should be low i.e. user’s request must be handled in less time.

2. Determination of Workload: calculating number of average user

3. Scalability: System should be scalable to handle large number of user request

4. The software should be able to handle at least 10000 users at a time.

5. Data should be secured and backed up every quarter hour.

6. Power supply should have a back up and a disaster recovery plan.

7. System should be operable 24 hours a day and accessible in real-time.

* 1. **Safety Requirements**

The Database may get crashed or damaged due to some viruses or operating system requirements. Therefore it is mandatory to have backup for your data.Ups/inverter facility should be there in case of power failure.

* 1. **Security Requirements**
* System will use secure Database.
* Proper user Authentication Will be provided.
* There should be separate account for Admin & user. So that no one else can access the database except Admin.
  1. **Software Quality Attributes**

Availability: Checking that the system always has something to function and always pop up error messages in case of component failure. In that case the error messages appear when something goes wrong so to prevail availability problems.

Usability: Checking that the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

Functionality: Checking that the system provide the right tools for editing question databases, creating

session tests and analysing the test sessions. In that case the tools that the Database editor provide are

the ones that provide that attribute.

* 1. **Business Rules**

User can chat with bot and schedule a call. Administrator can see the user credentials, see the chat and reply to it.

1. **Other Requirements**

We are required to collect the data of various laptops and mobile. Also we are required to analyse the data in order to give the best possible output to the user.

**Appendix B: Analysis Models**

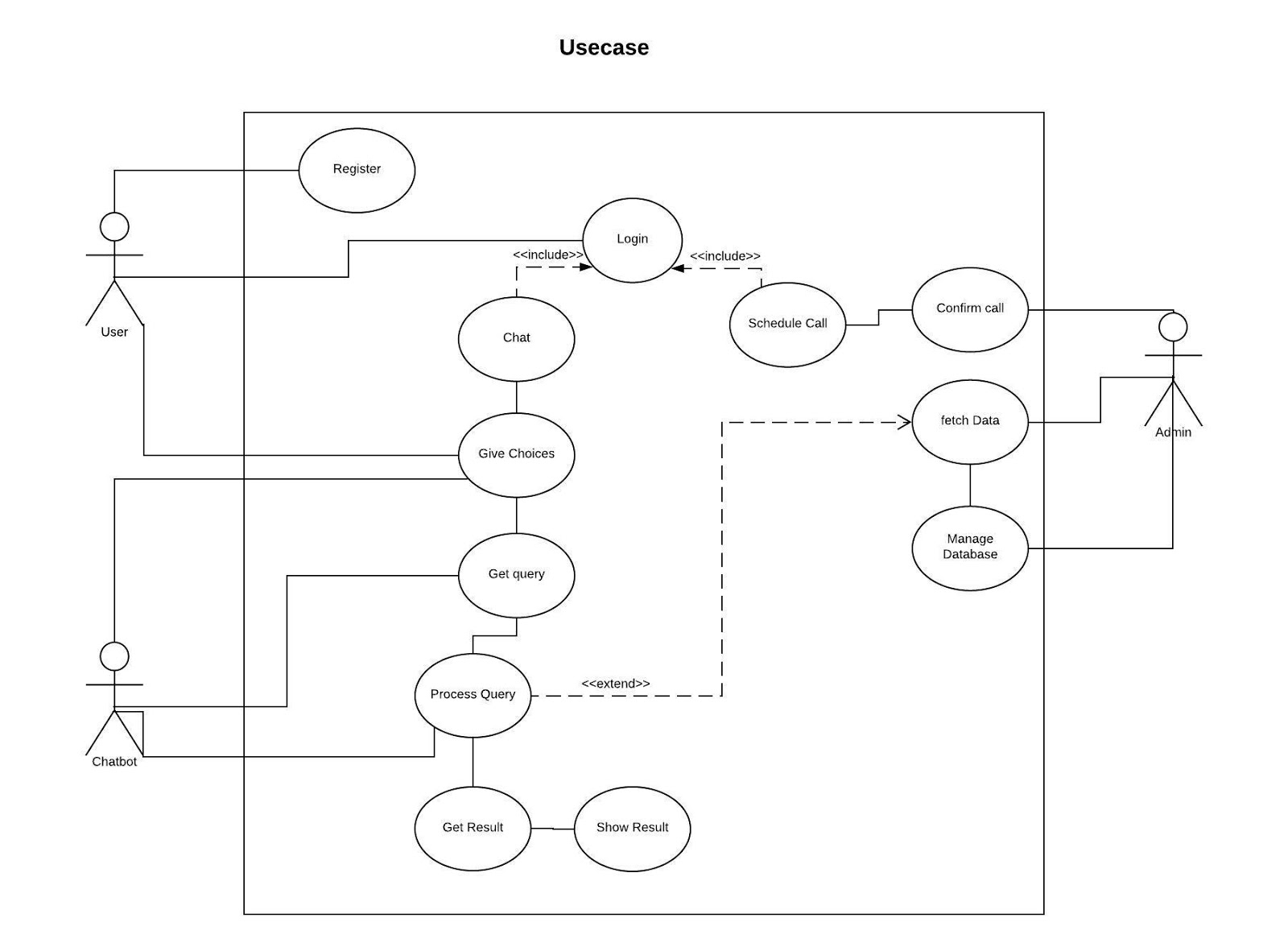


Fig 1.0 Use Case Diagram

**Description of Use Case Diagram**

**Actors**

* **Primary Actor**

**User:** A user is a primary actor which interacts with the system, provides inputs to the chatbot application and retrieves some output in turn of those inputs.

* **Supporting Actor**

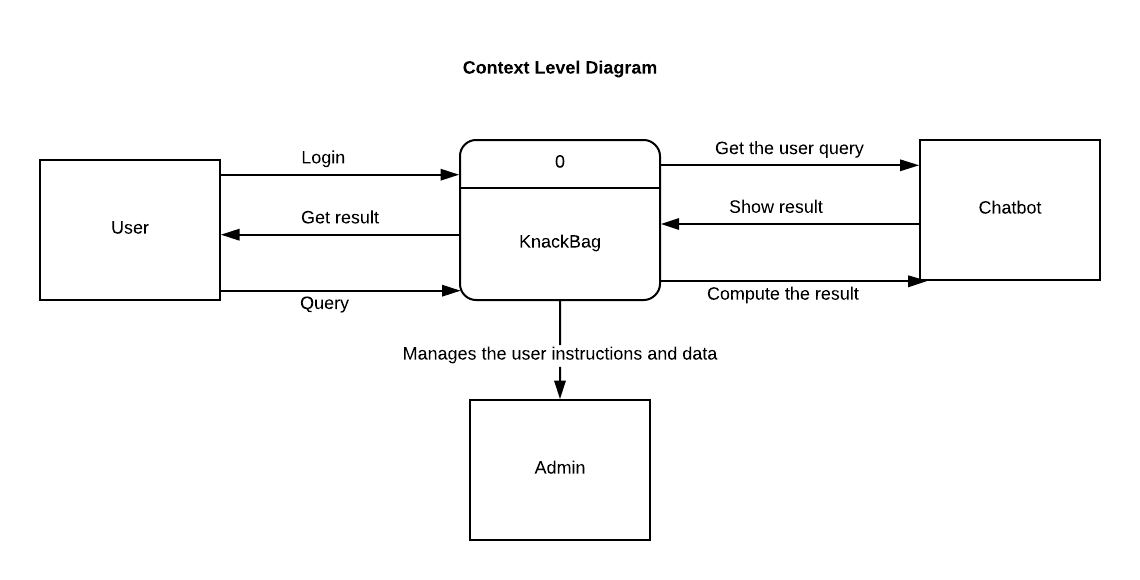
**Chatbot:** chatbot takes inputs from the user and on the basis of them it gives user the best result, it interacts with the user with text, buttons and list of options, users can choose in order to get appropriate output from the system.

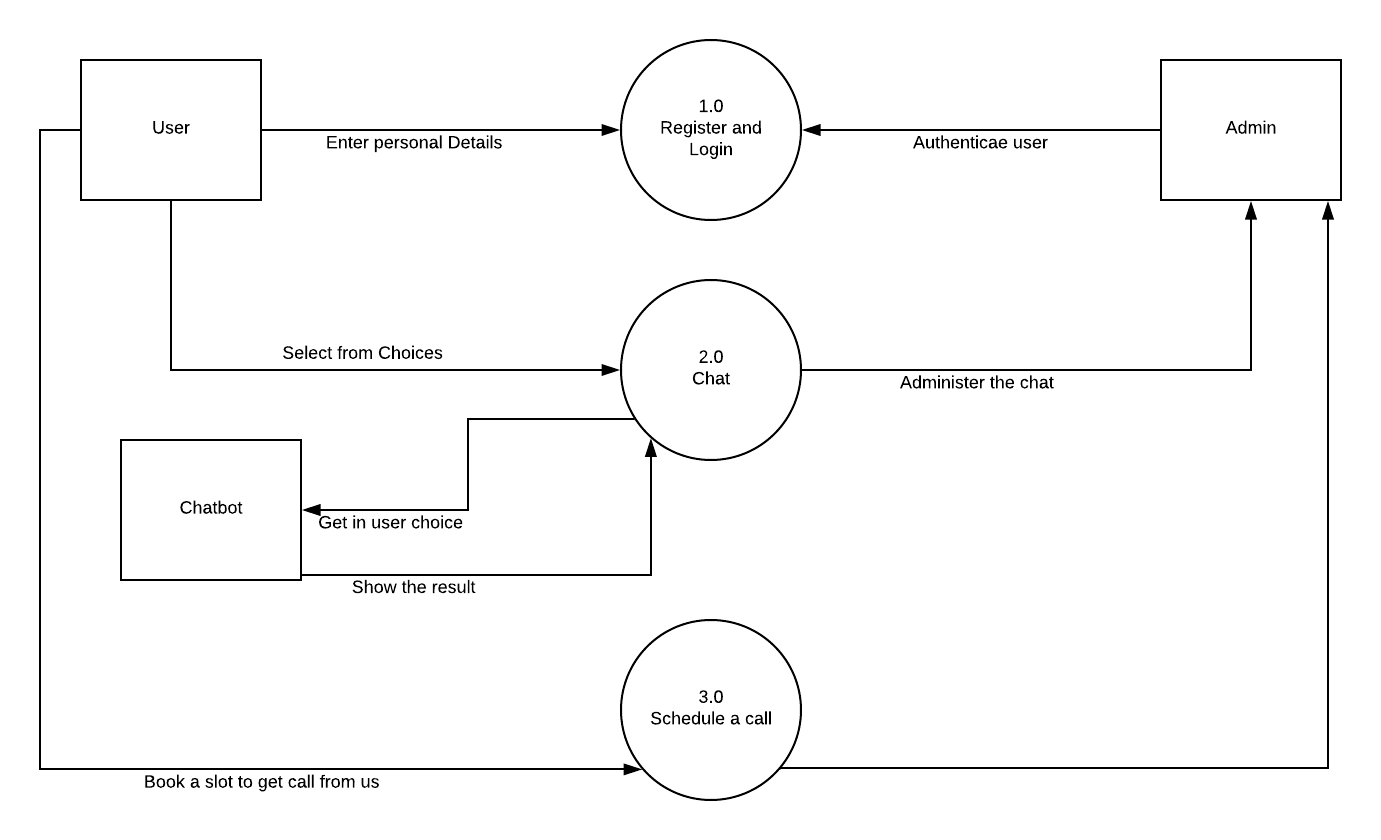
* **Offstage Actor**

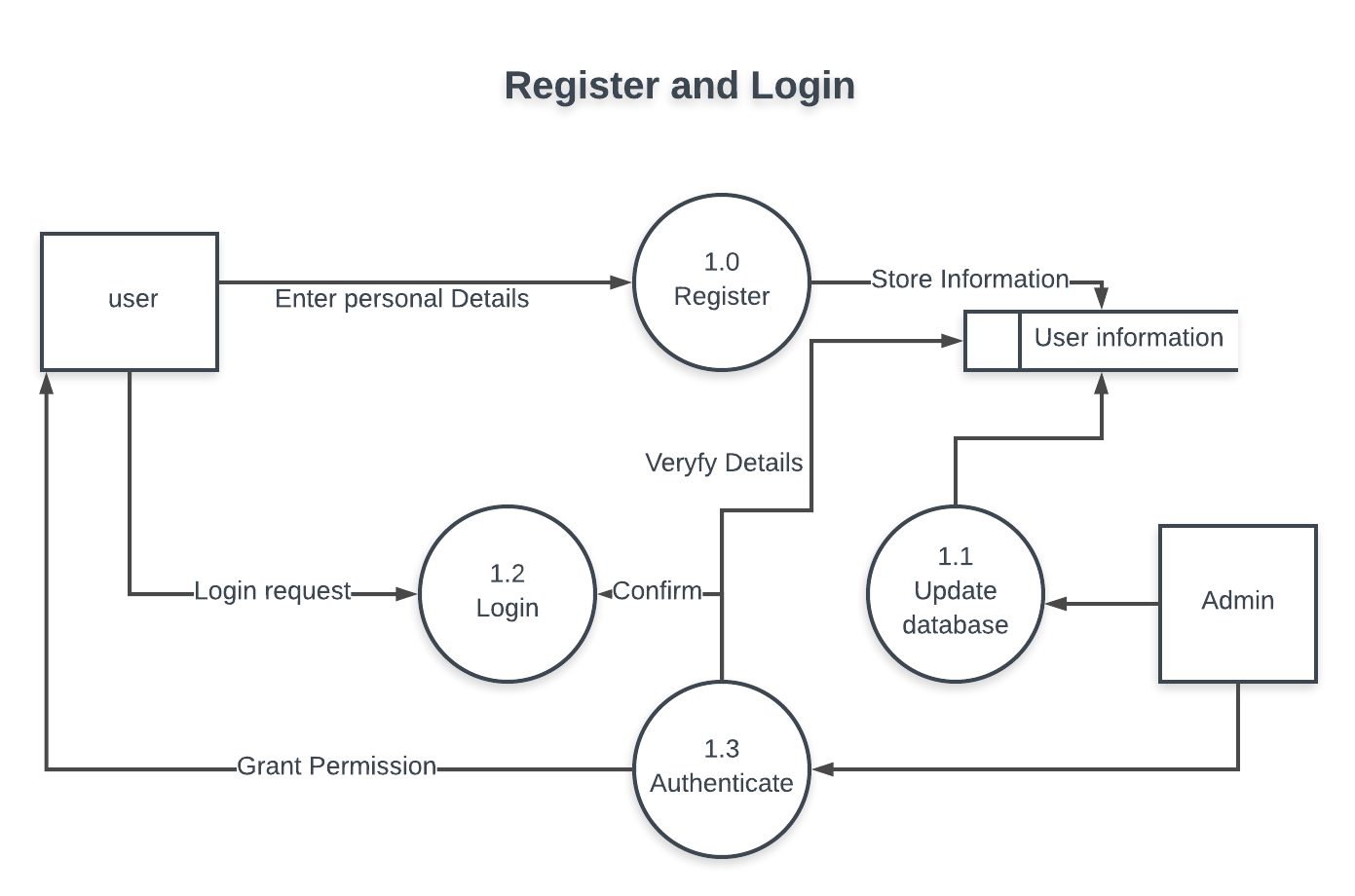
**Admin:** Admin is responsible for the processing queries taken by the chatbot from the user, retrieving best possible combination from database, maintaining database such as adding product details, modifying and deleting them is done by Admin.

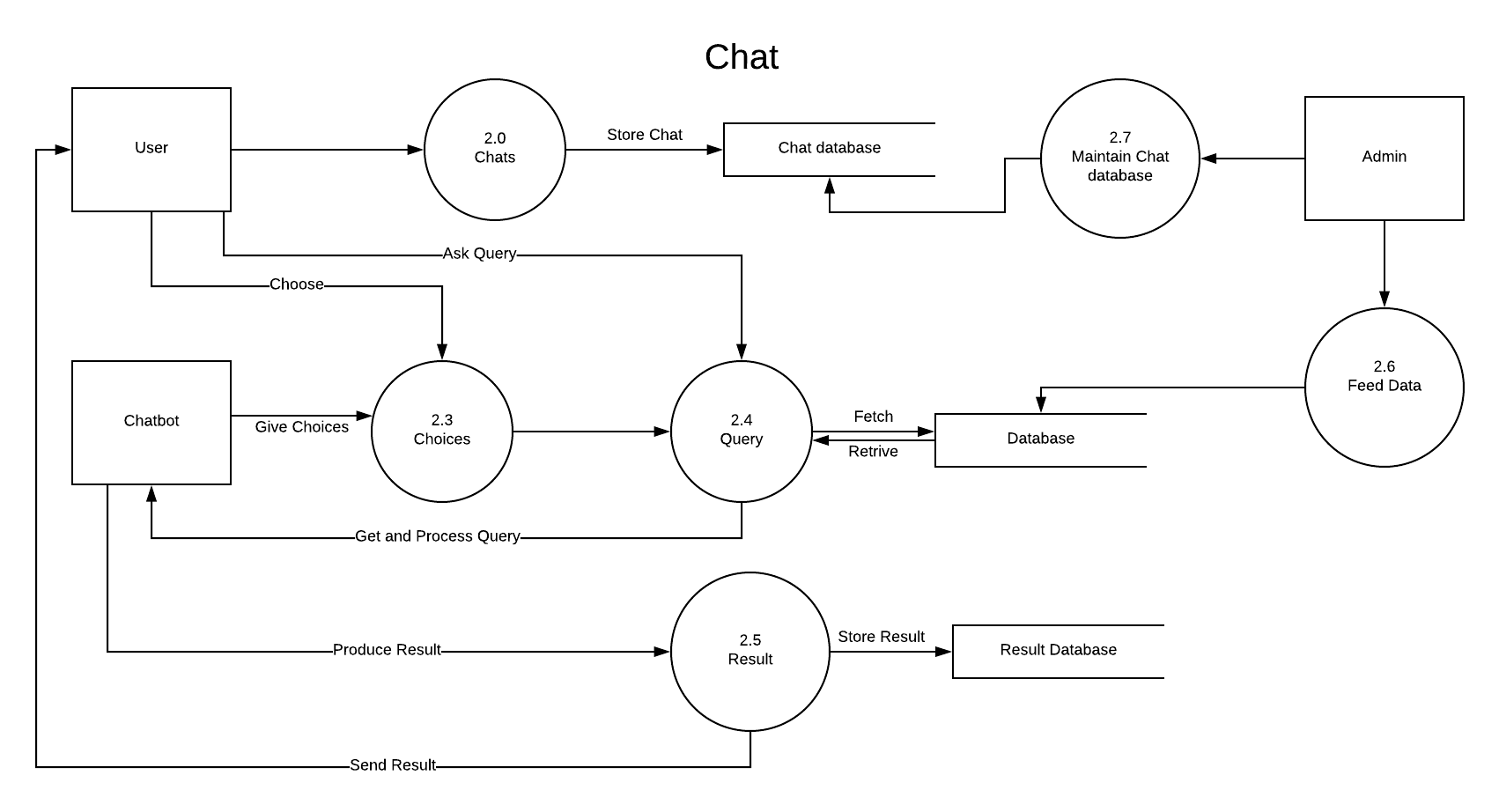
**Use cases**

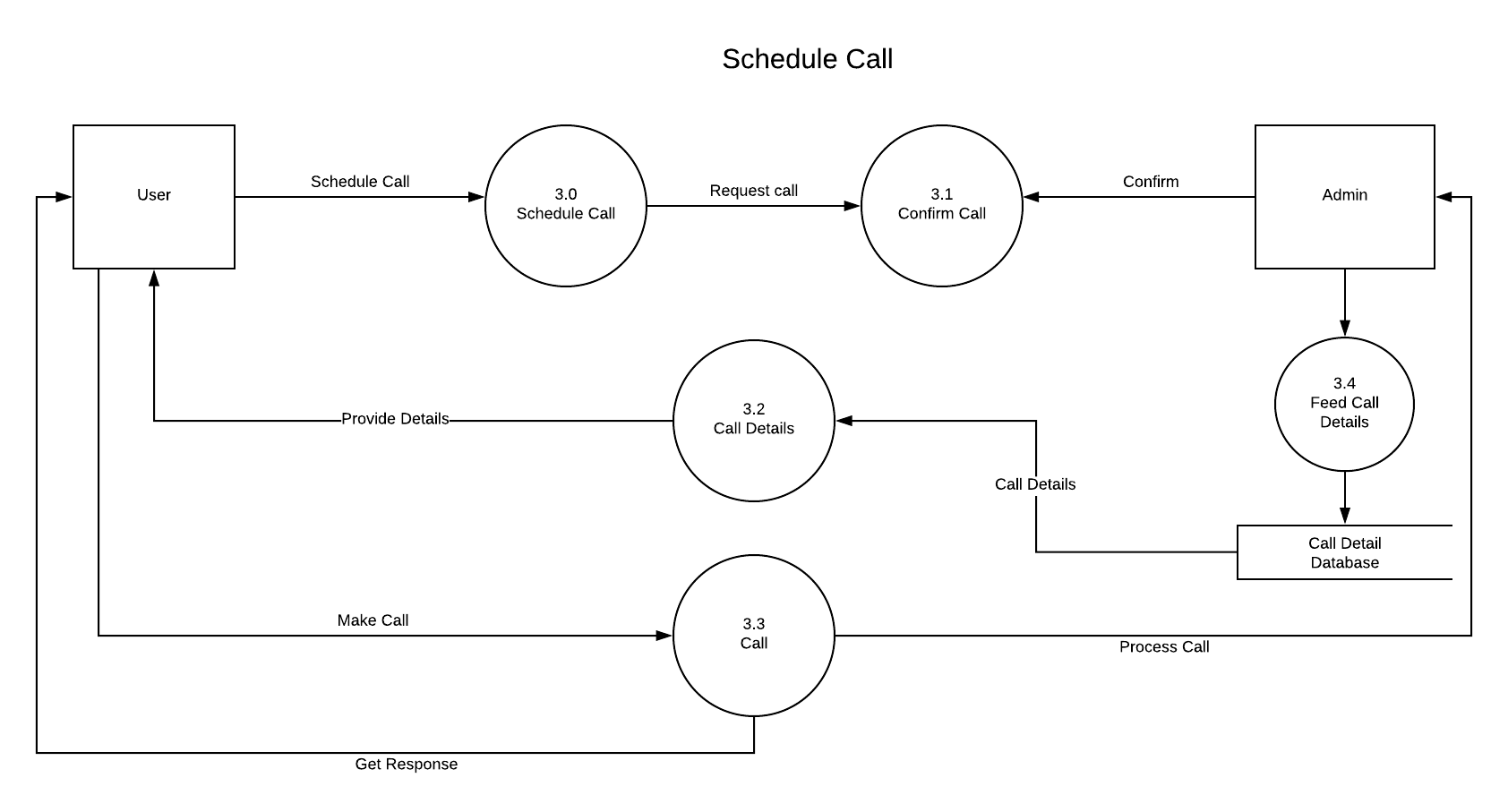
* **Register:** In order to interact with the system and use it user should be registered, this use case is for registering a new user with user details such as name, contact number, Email Id and password.
* **Login:** User must login with correct credentials in order to use system. In this use case user login with registered email-id and password.
* **Schedule Call:** User can schedule a call to enquire more about the products and any other query, for this user must be logged in.
* **Confirm Call:** Admin will confirm Scheduled Call from the user providing details such as date/time of call.
* **Chat:** User will chat with chatbot with the help of text and buttons provided in the interface.
* **Give Choices:** Choices in from of buttons will be provide to user from interface with the help of chatbot application.
* **Get Query:** When user will select its choice from the list of choices they will be accepted by the chatbot.
* **Process Query:**  The query is then processed by the chatbot, query can be general or database specific i.e. either chatbot can answer it explicitly without using database or queries which require some database interaction are given to admin.
* **Fetching Data:** Appropriate meaningful information from database is extracted from the database with the help of query.
* **Get result:** Result corresponding to query from database is received by the chatbot.
* **Show result:** Output is given to users in turn of options and queries given by them.
* **Manage Database:** Admin manages database. such as adding product details, modifying and deleting them, addition of more functionality is also performed by Admin.

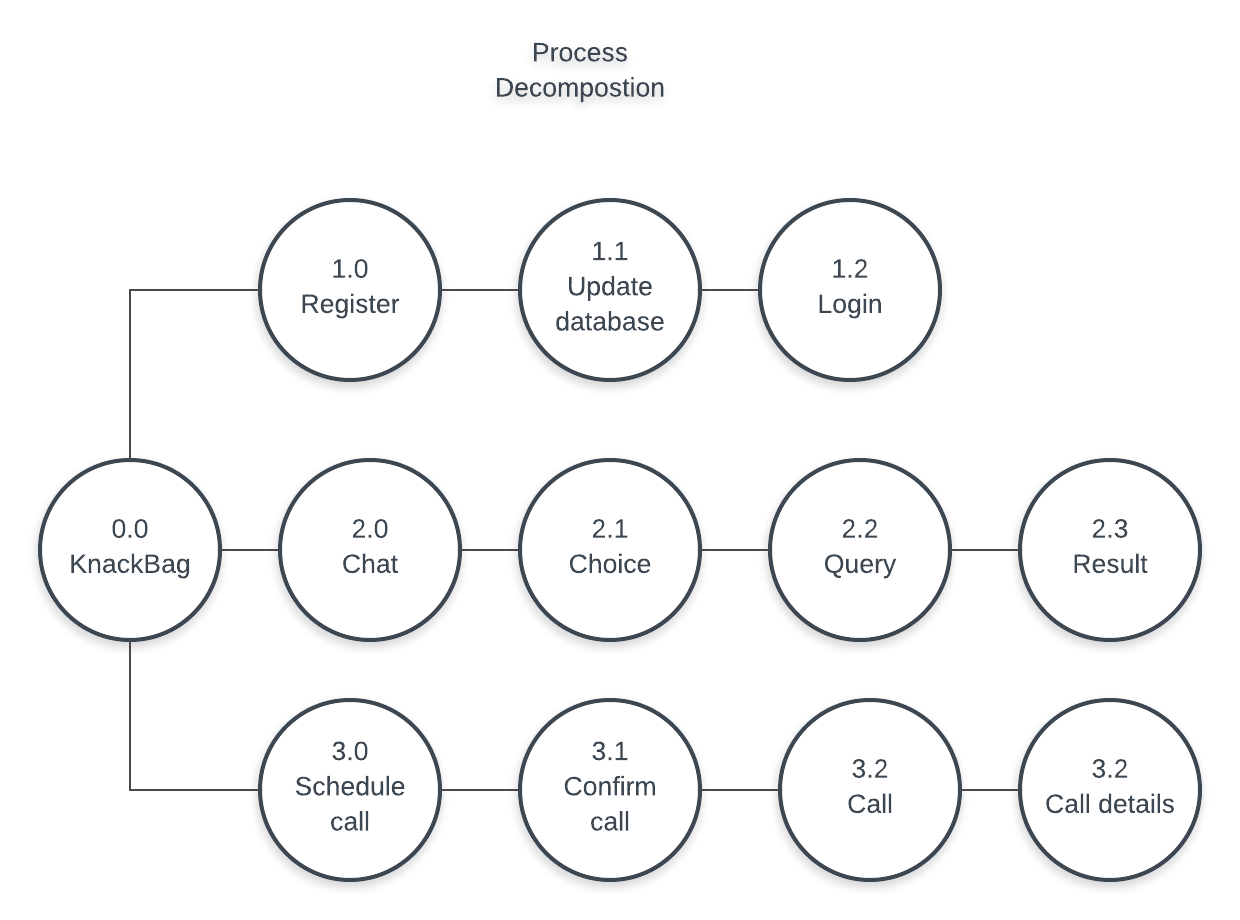
**DFD**  
  


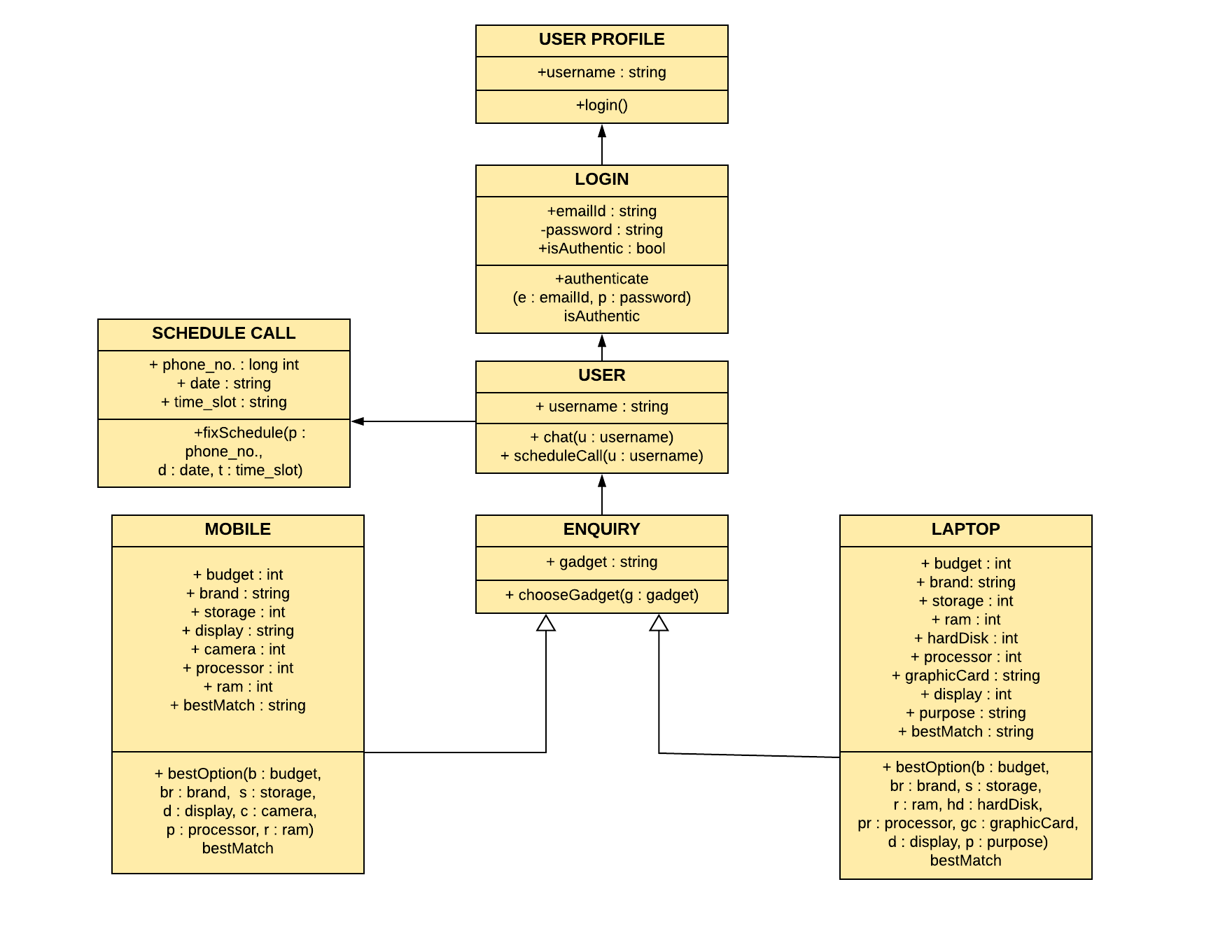
****

****

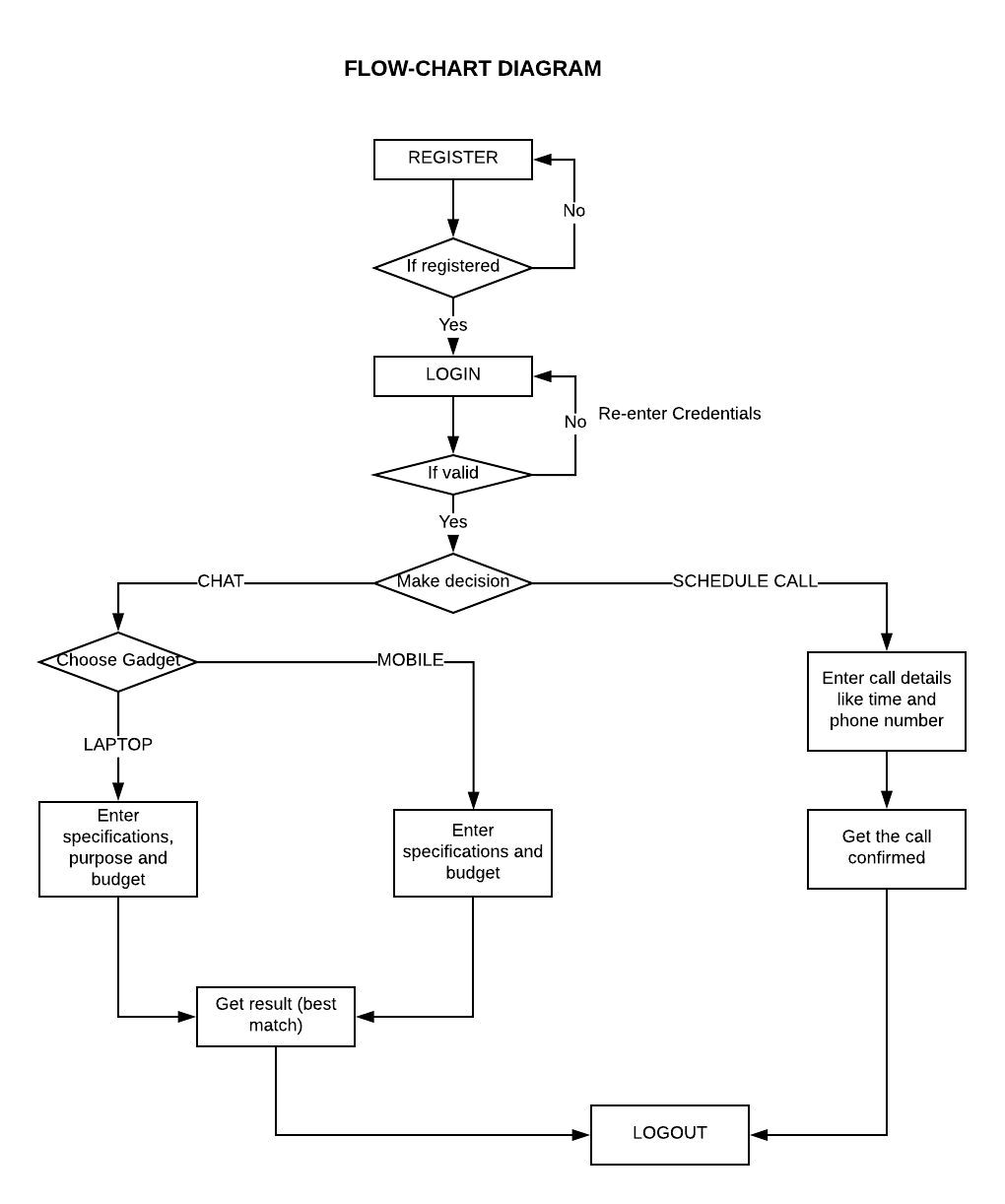
****

****

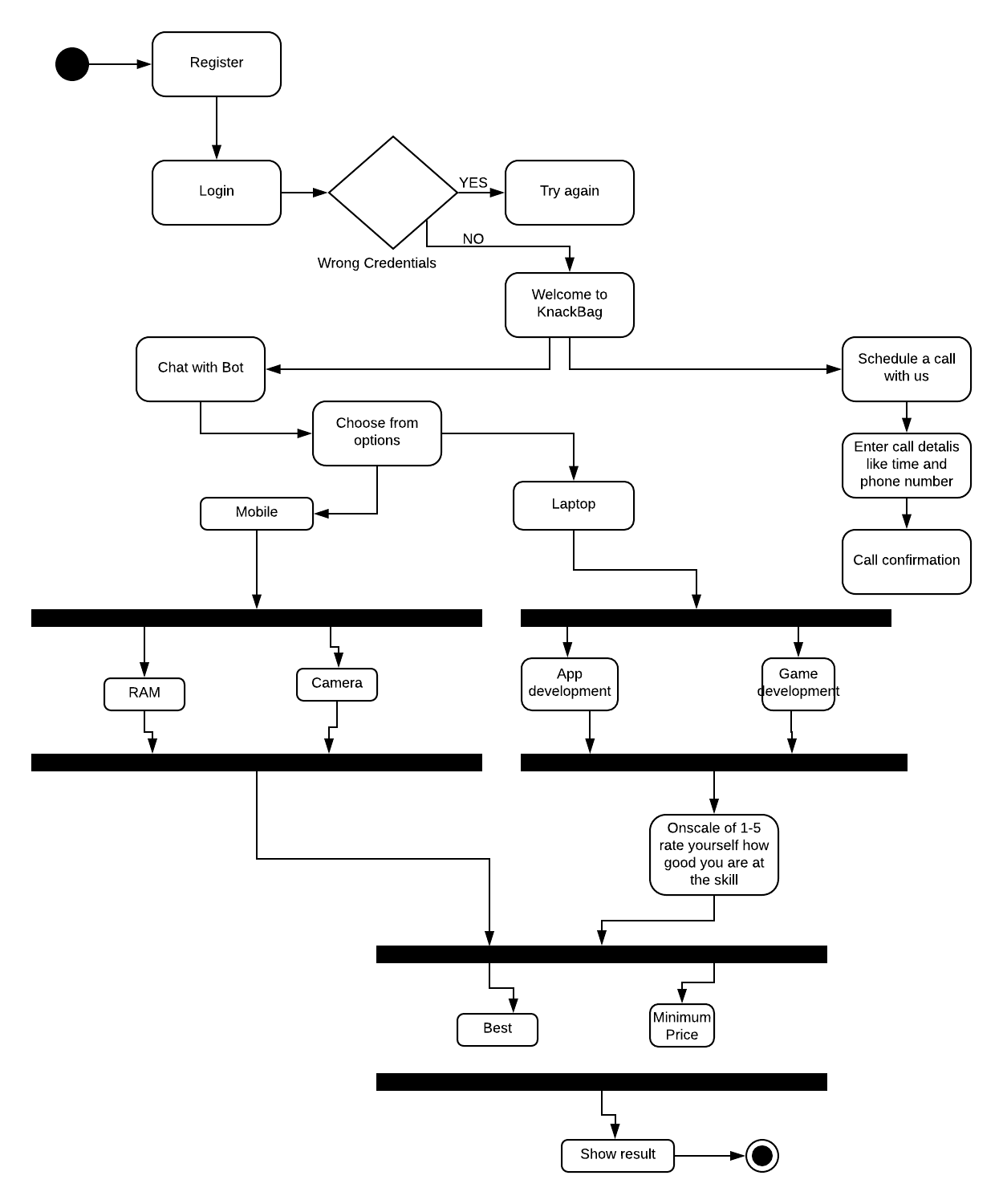
****



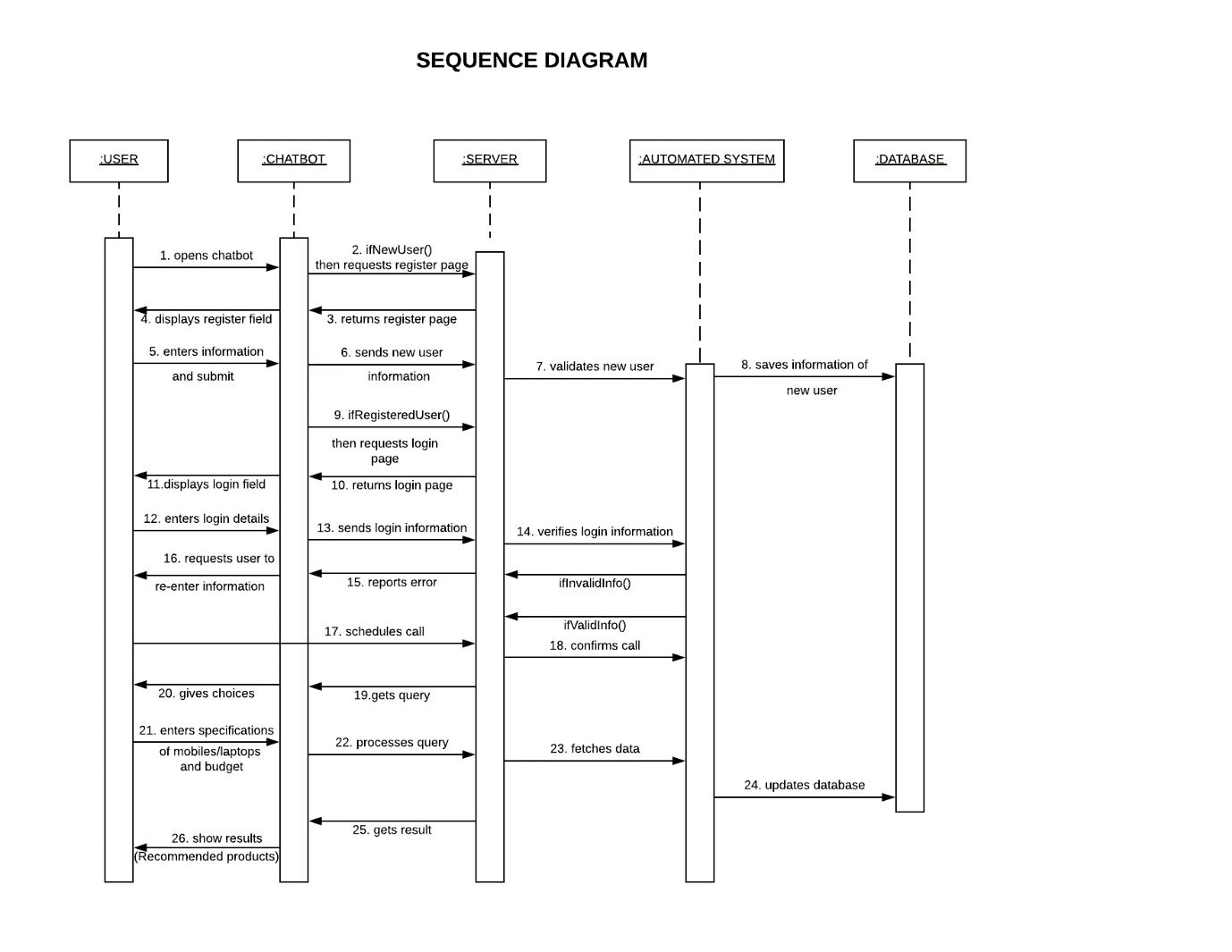
Class Diagram



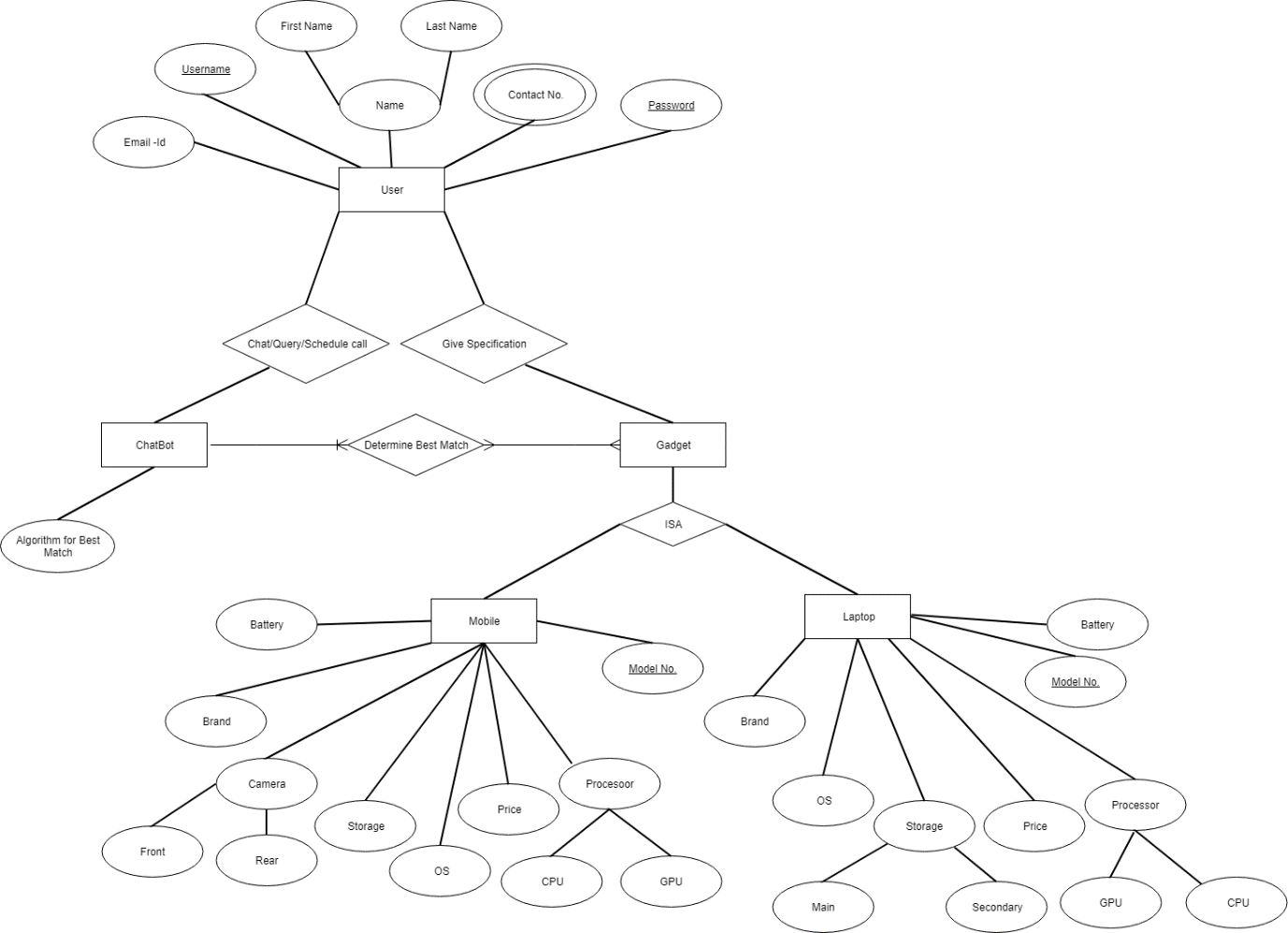
**Flow Chart**



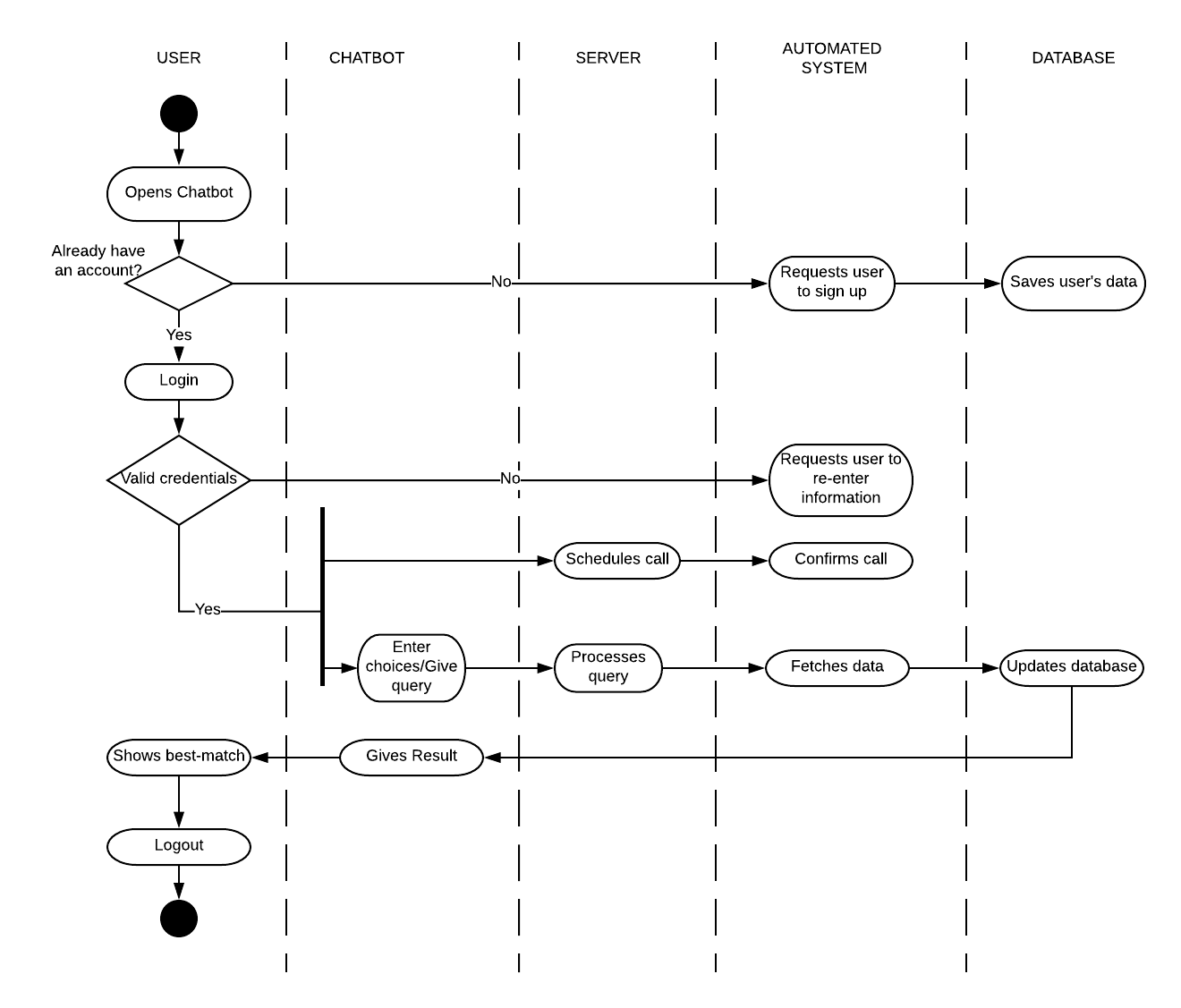
State Transition



Sequence Diagram



Entity Relationship Diagram



Activity Diagram