**Project ‘iDoctor’**

*By group Cyclopentane*

1. **MEMBERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **English Name** | **UCD num** | **BJUT num** | **UCD Email** |
| **Cai Yihua** | **Louis** | **15206083** | **15372224** | **yihua.cai@ucdconnect.ie** |
| **Chen Yixiao** | **Oliver** | **15205899** | **15372135** | **yixiao.chen@ucdconnect.ie** |
| **Du Wei** | **Charles** | **15205901** | **15372203** | **wei.du@ucdconnect.ie** |
| **Li Yucheng** | **Nathaniel** | **15205914** | **15372228** | **yucheng.li@ucdconnect.ie** |
| **Sun Li** | **Marc** | **15205925** | **15372229** | **li.sun@ucdconnect.ie** |

(Table 0.1)

1. **GROUP NAME & SYMBOL**

Our group name is ‘cyclopentane’. It is an organic molecule formed by five carbon atoms each of which stands for one group member.

Symbol of our group is a pentagon:

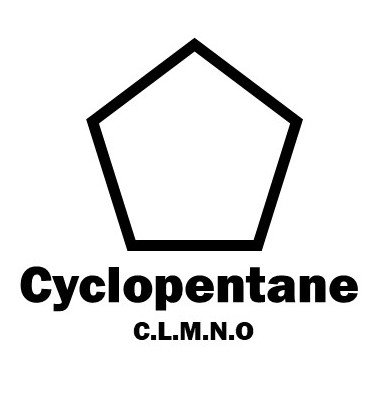


Figure 1.1

(C.L.M.N.O is the combination of all our English names’ initials)

1. **PROJECT SUMMARY**

Though it seems that the convenience of internet has swept our lives in every aspect, application we made specialized in medicine is not widespread as we think. Aiming at solving this phenomenon our project is about to make a change: ‘iDoctor’ designed to provide everyone a platform where you can make appointment with medical expert without waiting in line at hospital, meet people that suffers from the same disease as you do without fearing being alone, post your story to gain other people’s help & suggestion without finding no responses. ‘iDoctor’ will be operating as a website. Also, we will release android version for mobile users. Detail introduction of our functions are as follows:

* 1. Registration & social system

Internet can bring the all the information you want to you but it can not ensure those pieces of knowledge to be true. So the most important function iDoctor should have is allowing patients to make appointment with medical experts. Appointment can take two forms: 1st with doctors at public hospital, which is cheaper but can consume longer time. 2nd with private doctors which will be a little expensive but more flexible. This function, should it be able to run, need a chat room which should connect two individuals, also, it means that doctors should log in as a different identity, certificated by website administrators. Chat room should be placed in the website as a web application, clients’ contacts should be able to be stored in relational databases.

* + 1. Identity

This evolves with storing information in database. We are going to set up a table to store basic information about clients. One specific attribute will be assigned to store whether the client is a patient or a doctor. Based on what we are capable of; we are going to manage database by my SQL.

* + 1. Relations

Information about one client’s contact should be stored in the form of database. Considering keeping one database for each individual is a huge burden for the server, we decided to keep contact database as local file. This file will be referenced every time a client try to make a dialog.

* + 1. Chat Room

Expecting function of chat room is rather simple, we think using socket may be a good choice. But there is some question requires a closer look: like how to send message to an offline client. Clients can only talk to those who are already contacts with them, in other words: already exist on their contacts table.

* 1. Medical forum system

In order to allow users, both patients and doctors, to share their stories and experience conveniently, we plan to add a blog-based forum as the platform for users to edit, share, view and even favorite articles they like. To achieve this, we need to set up some prepared framework like word press.

* 1. Mobile version

It is important for our clients that they can browse our set anytime anywhere they want. So a mobile version is in need. Once we are finished with desktop version, we can wrap it in the form of android. It will not be hard to do.

* 1. Administrator

A good forum needs management. To make that happen, there should be administrators. They should be able to release notification, delete unregulated posts or advertisements that would harm forum’s environment, ban certain users who are keeping violating rules. Administrator have direct access of the back end of our website.

* 1. Sign up & sign in

New users should be able to use email to sign up a new account, then this email address will be stored in the database on the server. When they log in the second time, they can use that email address as user name.

1. **REQUIREMENT ANALYSIS**

Expecting functions mentioned above will cover all 3 requirement fields.

* 1. Networking

As we are talking about a blog-based online forum, that means there will be a main node that coordinate the rest. So our best choice is server-client mode. We plan to rent a server where we can set up managing software and database. One server will stay in charge of the rest, signing up, logging in, posting articles, adding friends, will all be under supervision of the server. Many processes will need info like whether a user have the permission of certain action, whether an account is on the contact table of a user so the user can dialog with it, etc. However, when connection for dialog set up between two clients, it is likely that they are proceeding the process in the frame of p2p.

* 1. Storage

Storage is the key of the whole system: as every account’s detail info should be stored in server’s database. Relational database will be our first choice, like my SQL. We are going to set up L.A.M.P on server and back end should be able to alter that database. Also, for dialog, we are going to store them into local files.

* 1. Front & back end interface

Front end service is where users operate, back end interface is where administrators going to operate. They have completely different limits of authority and using purposes. Clients can use basic functions like posting articles or adding friends. That will appear in the form of html webpages on desktop and app on android. Administrators have back end interfaces to monitor, control, modify the key data. Backend interface will only be on webpage. Admin should use special accounts to log in and operating.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT / GROUP NAME** | **Group Cyclopentane** | | | |
| **Start Date** | 17-03-2018 | **Finish Date** | 26-05-2018 | |
| **Aim /**  **Objective** | Use Django frame in Python to configure the server to hold the connection within different client. | | | |
| **Work package**  **Manager** | Charles(15205901) | | | |
| **Contributors to this package** | Oliver(15205899) | | | |
| **Description /**  **Activities** | Task 2.1 Rent the server Online and configure the environment, implement Python and Django in to the server. Make sure the client and the server can connection.  Task 2.2 Realize the online register function between the server and the web application. The user can online register on web page. This process need user to log in firstly. (Enter some basic personal information and the server can construct the patient medical report.) The patient’s medical report can only change by doctor with the patient’s authorize.  2.2.1 Realize the manage client (via web browser). To allow the hospital manage the register via management client. (e.g. time, patient’s information doctor)  Task 2.3 Achieve the online blog function, allow the patient client and the expert client to send personal blog to the server. Patient can upload their experienced and doctor can also do the same. To help the patient find the similar situation.  The search function should be implemented to let the patient query the similar illness.  2.3.1 Achieve the web browser – server blog function.  2 .3.2 Achieve the android – server blog function.  Task 2.4 Achieve the customer-doctor chat function. Patient client can chat with doctor. The patients can consult their status with doctor to achieve online treatment.  Task 2.5 Testing the whole system function coordinate with the front client. | | | |
| **Milestones** |  | | | Week |
| M 2.1 Configure the environment completely and the app can communicate with server  M 2.2 The user can register online  M 2.3 Any client can send blog(text) to the server and be displayed in webpage  M 2.4 Patient client can contact the doctor client  M 2.5 Test the whole system can realize the whole system completely  M 2.6 Final Release | | | **5**  **7**  **9**  **11**  **13**  **14** |
| **Deliverables** |  | | | Week |
| D 2.1  The server can hold the patient client to register online and hold any categories of client can send blog to server.  D2.2  The server can hold the chat function between multiple client. | | | **9**  **13** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT / GROUP NAME** | **Group Cyclopentane** | | | |
| **Start Date** | 17-03-2018 | **Finish Date** | 26-05-2018 | |
| **Aim /**  **Objective** | Use Django frame in Python to configure the management to hold the connection with server and client.  Use MySQLdb to save doctor and client’s information | | | |
| **Work package**  **Manager** | Oliver(15205899) | | | |
| **Contributors to this package** | Charles (15205901)  Louis (15206083) | | | |
| **Description /**  **Activities** | Task 1.1 Rent the server Online and configure the environment, implement Python and Django in to the management server. Make sure the management, client and the server can connection.  Task1.1.1 Establish a database connection with python and make sure data can be saved in database.  Task 1.2 Realize the online register function between the server and the web application. The user can online register on web page. This process need user to log in firstly. All information is saved in MySQLdb.  Task 1.2.1 Use management sever that allow the hospital manage the register via management client. All register information is saved in database. (e.g. time, patient’s information doctor)  Task 1.3 Achieve using management to manage the online blog function, allow manager can delete or change blog which send by client or doctor.  Task 1.3.1 Achieve the web browser – manager blog function.  Task 1.4 Testing the whole system function coordinate with the management, front client and server. | | | |
| **Milestones** |  | | | Week |
| M 1.1 Make sure the management, client and the server can connection  M 1.2 Establish a database connection  M 1.3 save register information in database  M 1.4 manager can delete or change blog  M 1.5 Final Release | | | **6**  **8**  **9**  **12**  **14** |
| **Deliverables** |  | | | Week |
| D 1.1 A management can connection with server and front client, and information can be saved in the database.  D 1.2 Management can change and delete blog, it also can release some announcements. | | | **9**  **14** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT / GROUP NAME** | **Group 3** | | | |
| **Start Date** | Week3 | **Finish Date** | Week13 | |
| **Aim /**  **Objective** | Work with my teammates to finish the subject of the Android system | | | |
| **Work package**  **Manager** | SunLi | | | |
| **Contributors to this package** | SunLi, CaiyiHua, ChenYixiao | | | |
| **Description /**  **Activities** | Task 3.1 Build a local database to store the privacy message to the Android end, and at this time we should have already built the database of the service end, so in this task we will try to connect the local database to the service one.  Task 3.2 translate some functions of html we have already finished to our mobile version.   * 3.2.1 sign up and sign in system * 3.2.2 relations chat room, so customers can find their friends and communicate with them. * 3.2.3 medical forum system, this is based on the finished service we have already built * 3.2.4 administrator system, so we can control the forum system easily and forbid the hostile attack   Task 3.3 Test if the app can work perfectly and the surface of app has a cool appearance. We even want to upload the app to Android app market and do a survey to get some suggestions about solving problems and bugs, so we can make our production more fashionable and convenience | | | |
| **Milestones** |  | | | Week |
| M 3.1 build a local database that can connect to the service end  M 3.2 basic model of application on the mobile version  M 3.3 finish the functions  M 3.4 optimize interface  M 3.5 finish testing  M 3.6 Final Release | | | **5**  **7**  **10**  **11**  **13** |
| **Deliverables** |  | | | Week |
| D 3.1 application with local databases that can connect to the service end  D 3.2 the finished application and result of survey | | | **8**  **12** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT / GROUP NAME** | **Group 3** | | | |
| **Start Date** | 2018/3/17 | **Finish Date** | 2018/5/26 | |
| **Aim /**  **Objective** | 1 Design the interface of the website  2 The front-end production and beautification of the web page | | | |
| **Work package**  **Manager** | LI YUCHENG | | | |
| **Contributors to this package** | CAI YI HUA | | | |
| **Description /**  **Activities** | Task 4.1 Building a basic web page framework with HTML (for example, the interface of users can register, log in, log out and leave messages)  Task 4.2 using css and div layout to beautiful the website (for example, add some doctor’s picture and hospital’s picture to attract users to visit) bootstrap will be used to construct the css framework.  Task 4.3 Using the JQuery and JavaScript to make the website Interactivity   * 4.3.1 there are three buttons in the enrolment interface, when users click the register button, then they will go into a new interface that users can register a new user and store the basic information in the database, when users have registered, then they can into the main interface and enjoy the all function in the website like consult some questions to doctor. After users want to exit the website, they can click the log out button, then the information will be wiped out from the website) * 4.3.2 After the users successfully landed on the website, they can use the all functions in the website, like when you click leave messages you can chat with doctors. And when you input the healthy problems then you can get some information about it. And when you click the picture of the doctors, you can get the detailed information about doctors.   Task 4.4 Testing the work package implementation(optimizing user experience and enriching functional modules, check the coexistence of web elements) | | | |
| **Milestones** |  | | | Week |
| M 4.1 basic web framework, there is a general shape of the doctor website, and have basic words and picture in the website  M 4.2 have a beautified website, there are some vivid typefaces and appealing pictures The website has better typesetting, make the website is appealing  M 4.3 have an Interactivity website, users can contact with doctors or acknowledge some basic information about the disease and chat with others who get a same disease  M 4.6 optimizing user experience and enriching functional modules, check the coexistence of web elements) | | | **5**  **7**  **11** |
| **Deliverables** |  | | | Week |
| D 4.1 some basic html and css language to construct the basic framework of the website  D 4.2 write some JavaScript and jQuery to achieve the interactivity of the website | | | **7**  **13** |
|  |  | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT / GROUP NAME** | **Group 3** | | | |
| **Start Date** | 17-03-2018 | **Finish Date** | 26-05-2018 | |
| **Aim /**  **Objective** | Log in & sign up function | | | |
| **Work package**  **Manager** | Cai Yihua(Louis) | | | |
| **Contributors to this package** | Chen Yixiao(Oliver) | | | |
| **Description /**  **Activities** | Task 5.1 allow users using email to sign up for an account  Task 5.2 allow users to use email as default user names to login  Task 5.3 store users’ account information in database  Task 5.4 test the whole system, see if a user can sign up or log in | | | |
| **Milestones** |  | | | Week |
| M 5.1 User can visit the website  M 5.2 User can sign up by email address and log in using that address  M 5.3 server can return an email when signing up  M 5.4 User can use a username to log in  M 5.5 Final Release | | | **7**  **9**  **10**  **13**  **15** |
| **Deliverables** |  | | | Week |
| D 5.1 Version 0.1 that users can sign up and log in  D 5.2 Version 0.2 server will return a certification email and user can log in with an alias they like  D 5.3 Final release | | | **9**  **13**  **15** |

1. **TIME LINE**

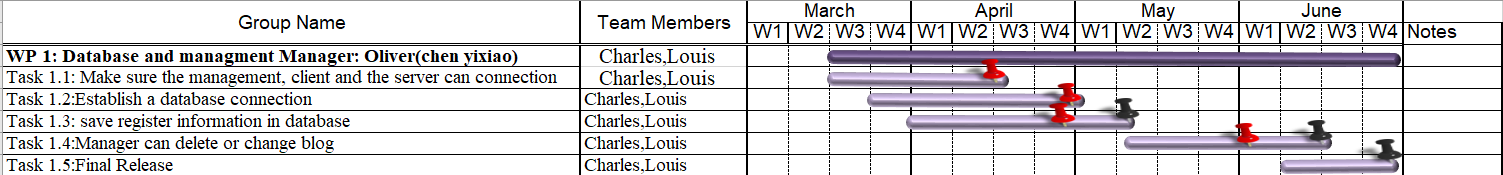


Figure 4.1

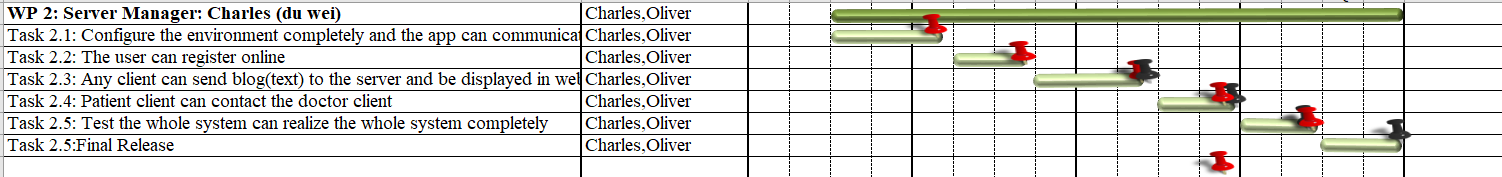


Figure 4.2

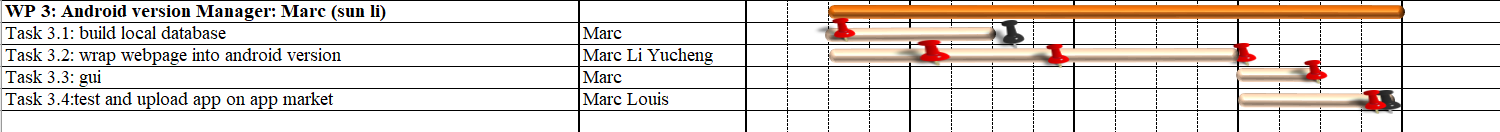


Figure 4.3

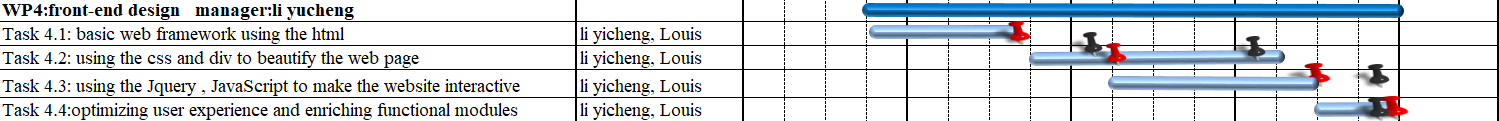


Figure 4.4

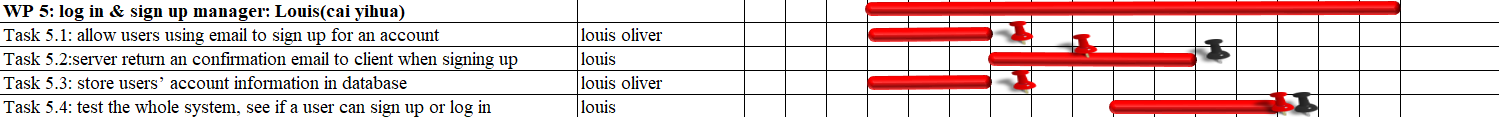


Figure 4.5