

Database Fundamentals and Mobile Computing

Assignment

Introduction

The Database Fundamentals and Mobile Computing assignment is a group assignment in which you have to create a database driven android application. Each group is assigned a different assignment from the set of assignments below. All projects will need some authentication system which allows a user to login using a password or register as a new user if they are not in the system, but other than that they are quite different. Note that where projects require different types of users, you can allow people to select their user type at signup (Even if that would not really make sense in the real world). We are deliberately keeping the project descriptions short as we would like to see what innovative interfaces, designs and workflows you come up with. For instance, we have not made it clear where user profile pages are required, etc. You should make these decisions to produce a good experience for your users. Mostly, you're way better at this than we are.

Note that you cannot select your project. A project will be assigned to you unless we have agreed to let you work on your own suggested project. Please email us to suggest a project idea that you'd like to work on instead of one of the existing ones.

How to create a group

On the moodle course for Mobile Computing (COMS2013), you will see a Group Management activity. You can click on this item to create and join groups. Note that if you are only doing DBF, and are thus not in MC, you should STILL do it through the Mobile Computing page on moodle as this allows us to keep a single central place for the groups. The maximum size of a group is 2, meaning that you can work by yourself or with one other person.

You must select a group by Monday, May 18th

1 Disease Tracker

This app is meant to help with contact tracing in the Covid-19 pandemic. The idea is that users will voluntarily "check in" at locations to indicate that they are currently at a particular location. The system will record all of these check ins. Users can then report if they are diagnosed with Covid-19, and if they are, any other user who is at risk can check this when they log in.

1. The system must have a predefined list of locations. (GPS support is not required but it will be very impressive if a group does include this)
2. Users must be able to indicate that they have been to a particular place on a particular day.
3. Users must be able to indicate that they have been diagnosed with Covid-19
4. Users must be able to query whether they are at risk for Covid-19 because they were at a particular location on the same day as someone with Covid-19.

5. Users must be able to see a list of locations along with number of infected people who have been to that location.

2 Anonymous Counselling System

The app provides a platform for people to access anonymous online counselling, which is increasingly important during the pandemic. The system must allow counsellors and users to join the system, and must match counsellors to users, allowing the counsellor to communicate to his assigned users.

1. Users must be able to indicate the problems they are in need of counselling for (Note that signups are anonymous)
2. Counsellors must be able to indicate the types of problems they are comfortable dealing with (This should correspond with the problems in the point above)
3. The system should assign a user to the counsellor that has the least users currently assigned and who matches the criteria regarding problems.
4. The system should have chat windows. Every user will only see one chat window as they only have one counsellor, but the counsellors should have a chat window per assigned user (Think of a menu system similar to the mobile chat programs you've used)

3 Lend a hand

This app allows users to help their fellow citizens. The app has a predefined list of resources that people might need, and connects people who have a surplus to people who have a shortage. For example, tinned fish or R12 airtime vouchers might be a resource. The app will allow any user to specify which items they are willing to donate, and how much they're willing to donate. Users will also be able to say what they're short of. The app will then match people so that a donation can take place.

1. Users should be able to sign up and select the number of items in each category they are willing to donate
2. Users should also be able to solicit donations of particular items, indicating the number needed
3. Users should be able to write a short biography of themselves as a motivation for items that they are in need of, and contact details so that the donation can actually happen
4. When a user indicates what they want to donate, the system will present a list of users who could use those items, and allow the user to allocate an amount of items to that person. This should decrease the amount that person needs as well as the amount the donor has available.
5. Users should be entered into a Donors list which is viewable by everyone, and ordered by the number of items donated.

4 Restaurant Order Tracker

This app is meant to inform users of the status of their orders at takeaways. In this app you must cater for multiple takeaways. This app is used by two distinct types of people - staff and customers. The app should offer the following functionality.

1. Staff should be able to add an order. The order doesn't need to record what items were ordered, but they must record who the order is for, which staff member created the order, and what time the order was created. Staff should also be able to see the average rating their orders have achieved.
2. Staff should be able to mark an order as either pending (which is the default state), ready (meaning that the order is waiting for the customer to pick it up) or collected (meaning that it has already been collected by the customer).
3. When a customer logs in, they should be able to see the status of all their orders, including which restaurant it is from. Customers should be able to rate the service received on each order with a simple thumbs up or thumbs down.

5 Remote Artisan App

One difficulty during lockdown is dealing with all the situations at home that would normally have you call a professional. If you need someone to help sort out your DSTV or your plumbing it can be difficult as you don't want to expose yourself to Covid-19. There are video tutorials on Youtube for solving most things but they don't have your particular situation in mind. The app should connect you to an artisan who is willing to do a consultation over a video chat platform in order to help you solve your problem.

1. Artisans should be able to sign up and list the expertise they have (from a predefined list)
2. Users should be able to search for artisans of a particular type, sorted by average rating
3. Users should be able to rate artisans they've used before, and should be able to add a comment with the rating
4. Users should be able to view the individual ratings and associated comments on artisans' profiles
5. Artisans should be able to rate users according to how pleasant the experience was, and comment on the rating
6. Artisans should be able to view the individual ratings and associated comments on users' profiles
7. Note, you do not have to set up a video chat or anything, just provide contact details so the parties involved can do it themselves.

6 Surrogate Shopper

The virus causes much higher death rates in at risk individuals such as the elderly. But these groups of people still need to purchase necessities. The app allows people to volunteer to do shopping for people who should not be out shopping themselves.

1. At risk individuals should be able to request items, with each item being a free text entry to allow flexibility
2. At risk individuals should be able to put their delivery location on their profile (You must use the phone's location services for this)
3. Volunteers should be able to view all requests, and select requests to view the details
4. Volunteers should be able to mark a request as taken, so that only one person ends up buying the items on the list
5. Volunteers should be able to view the requestor's location so that they can use something like google maps to navigate there
6. At risk individuals should be able to post messages of thanks on volunteers' profiles

Submission

You will submit the following items:

- Your code - you must upload your Java code, XML layouts and PHP files
- Demo video - you must create a short video (max 5 minutes) of your app in action. Consider using Open Broadcast Studio for this. Show the functionality and describe what is happening.
- Code video - you must highlight interesting sections in your DB design and java code (max 5 minutes)
- Project document - You must produce a document detailing some parts of your design (Details appear below)
- Compiled app - The app must be uploaded as an apk if using Android or a war file if using the android alternative

All of these must be uploaded on moodle for MC and Sakai for DBF by Saturday 20th June. Please feel free to contact me or Hima for any clarification or to suggest your own topic.

For DBF, the following characteristics will be important.

- A clear description of procedures/processes involved in the project/topic.
- Formulation of business rules (should be bidirectional).
- Development of initial ERD.
- Recognition of issues such as M: N relationships, NULL values, multivalued attributes, etc.

- Solutions provided and updated ERDs (Final ERDs should contain all the relevant information).
- Implementation of tables/functions/procedures/views/triggers need for the project.
- Demo video of the project/database.

For MC, the following characteristics will be important.

- The app must work!
- It must be able to send and receive data from the webservice
- Users must be able to register and login. Errors must be handled well.
- It must validate user input
- There must be multiple screens in the app and it should switch screens as necessary
- The correct data must be displayed and recorded according to the functional requirements
- The app must be visually appealing
- It must be bug free