Team: Black(bscs13024,bscs13064)
Project: SideKick
Assignment #1

SideKick is an interactive crowdsource product delivery application. SideKick will enable community members to receive and deliver products with ease and in the quickest way. SideKick will have two types of users - regular customers and sidekicks. Regular users are customers that will place orders using our SideKick service. This will help the users save their time and resources. Sidekicks are the users that will deliver the products to regular customers. This will help Sidekicks earn some extra money.

Requirements:

SideKick is an Android smartphone application, which will require an Android Studio for development, an android phone for testing and an internet connection.

Question #1:
Do you think you need mathematical verification of correctness of your system or a part of your system? Why?

Answer:
Yes, we need mathematical verification of correctness our system because our application relies on sending data between the closest devices in terms of GPS coordinates and highest user rating. For that, we need to mathematical verification of our algorithm to see if it is sufficient for our use.

Can you separate various concerns of your project from functional and quality perspectives? Highlight the concerns and describe how can you handle concerns separately?

Answer:
Functional concerns:

- Getting Login information from the users.
- Storing Login information to the database.
- Storing Orders to the database.
- Sending Order Request to the user.
- Get acceptance and rejection request, and deal accordingly.
- Send confirmation request back to the user

Quality perspectives:

- As little user clicks as possible

- No delays
- Errors are handled efficiently.

Question #3:
Identify some functional modules in your system. Discuss coupling and cohesion aspects.

Cohesion:

Each separate screen has different purpose. 1) Login screen. 2) User delivery location. 3) Target location. 4) Make a list of items screen. 5) Order class.

Coupling:

Each screen collects data for an instance of Order class and then that instance is uploaded to the database.

Question #4:
Identify the potential future changes in your system. Pick one potential change and discuss how would you address it in your system?

Answer:

If the user wants to order from more than one location then the system needs to be updated. A new class would be made that have a linked list of the instances of Order class.

Question #5:
Which increments would your suggest if you are asked to build your system incrementally?

Answer:

1. Get login
2. Get user location
3. Get target location
4. Get order list
5. Get order confirmation and place order
6. Upload data to database
7. Find the nearest user
8. Send acceptance request to the user