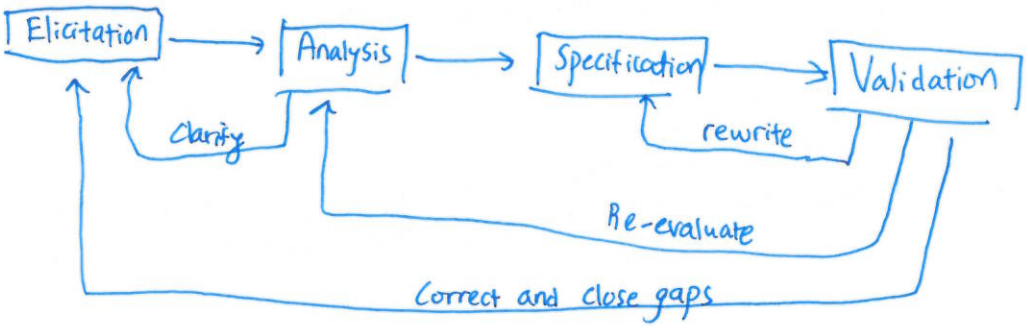
CZ 3003 Software System Analysis and Design AY 18|19 S1

Nathanael S Raj
nathanae001@e.ntu.edu.sg

1)(a)



(b) Listening Skills – The analyst needs to be able to pick out vital information from the Client and understand the needs of the project

Organisational skills – The analyst must be able to organise the elements of the project well and its dependencies

Interpersonal skills – The analyst needs to both manage his coworkers as well as his Clients well. Communication is key

(c) Watch users do their jobs – Useful for projects where you are replacing a manual function with an automated system. E.g. Online voting. Can watch manual voting process to determine the corresponding automatic functions.

Prototyping – Useful in many scenarios. E.g. can have a prototype mobile application for a iBanking app to receive feedback on the suitability of its design for various functions

Interview stakeholders – Useful especially in situations where it is uncertain what the exact demands of the project are. E.g. When creating a chat bot for a booking system, it is useful to interview both users and the management to understand the usual workflow.

(d)

Condition	Requirement Number			
	1	2	3	4
SG citizen | PR	F	T	T	T
Monthly salary > $3K	-	F	T	T
No Financial Issues	-	-	F	T
Action				
Accept Application				✓
Reject Application	✓	✓	✓	

Completeness – Yes. We have accounted for all possible permutations of conditions in this table.

Consistency – Yes. There are no contradicting requirements that lead to contradicting results.

2)(a) Inverse relationship. As we try to minimize the resource footprint of a software, it becomes less reusable for other applications. To optimize performance, the software must be tailored to fit the specific requirements of a project, reducing its reusability.

E.g. ~~Consider a search engine indexing technique. Words are often char~~ in a

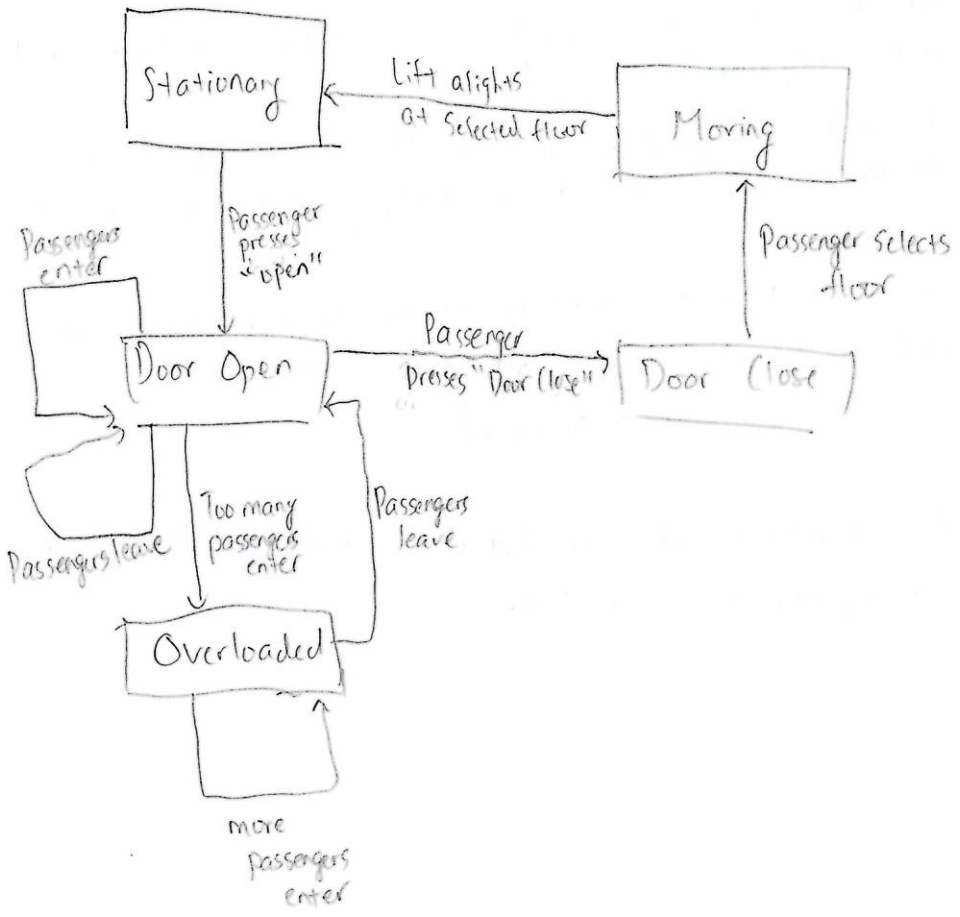
Consider storing timestamps in a database. To save space, if we knew that we were only interested in the date of the entries, we would not store the time. However ~~if we were to reuse the database for~~ this would reduce its reusability in a project where time needs to be stored.

(b) Low Coupling and High Cohesion - Low coupling means having few interconnections across modules in terms of both statically the links between subroutines of different modules and dynamically the frequency of subroutine calls aross module boundary at run-time. High cohesion means that there should be high interconnectiveness between subroutines of one module.

Separation of concerns - A complex problem can best be solved by initially devising an intermediate solution expresed in terms of simpler independent problems.
[from textbook]

(c) The filter transforms or filters the data it receives via pipes connected to it.
The pipe is the connector that passes data from one filter to next.
Pros - Simplicity, maintenance and reuse, concurrent execution.
Cons - Interactive transformations are difficult, no filter cooperation.

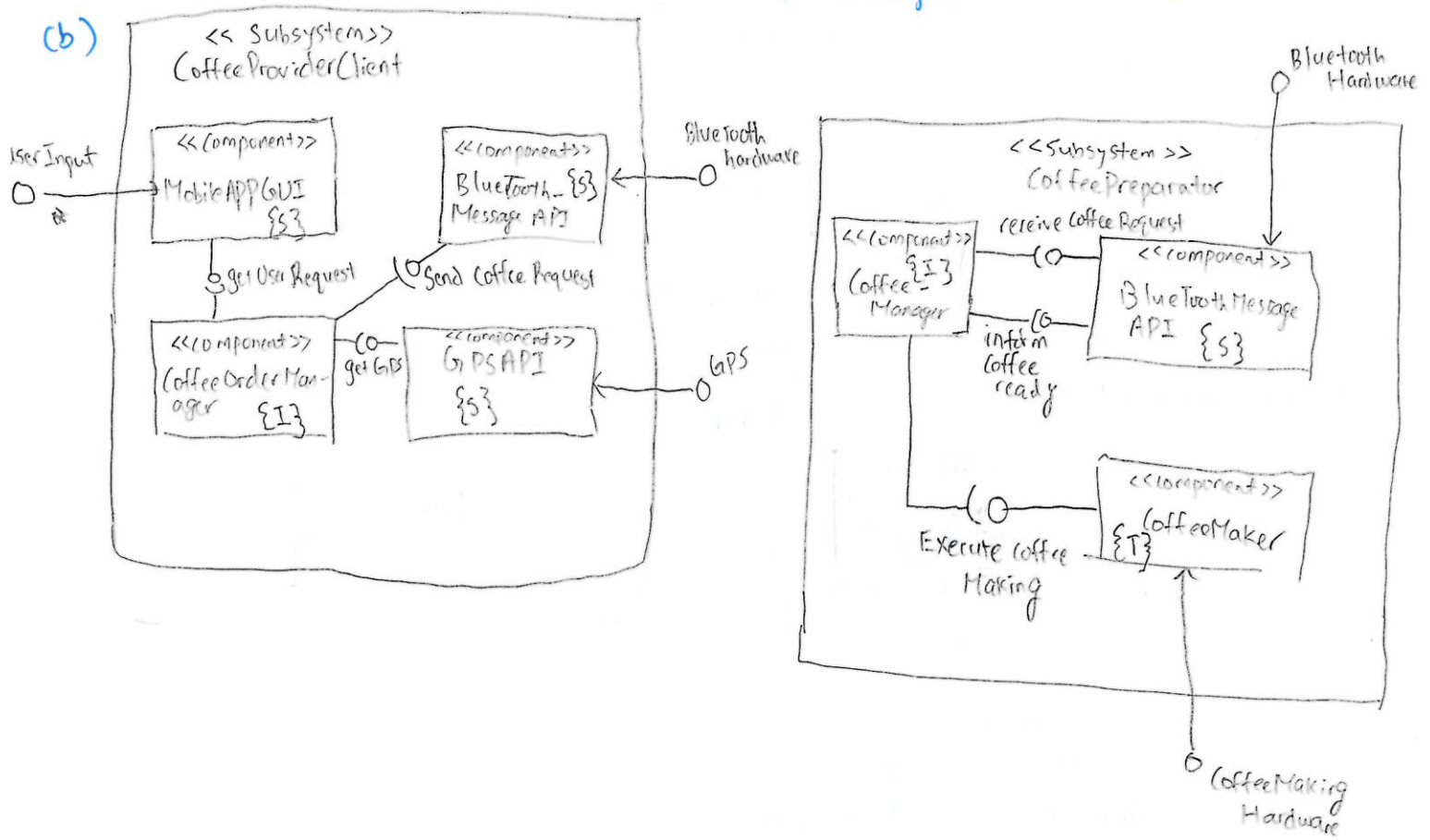
(d)

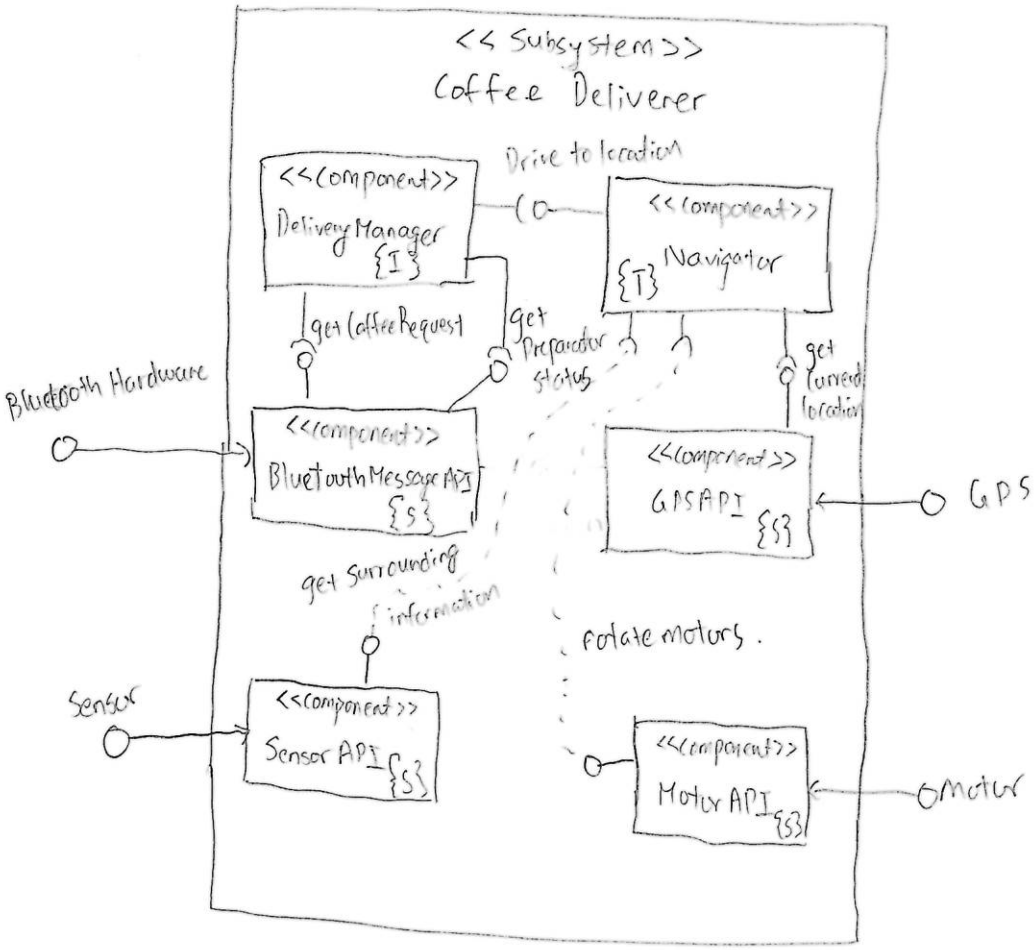


3) (a)

1. Coffee Provider Client
   - ↳ Bluetooth Message API
   - ↳ GPS API
   - ~~↳ My Preparator API~~
2. Coffee Preparator
   - ↳ Bluetooth Message API
   - ~~↳ My Deliverer API~~
   - ↳ Coffee Maker
   - ~~↳ Coffee Preparator Manager~~
3. Coffee Deliverer~~er~~
   - ↳ Bluetooth Message API
   - ↳ GPS API
   - ↳ Sensor API
   - ↳ Motor API

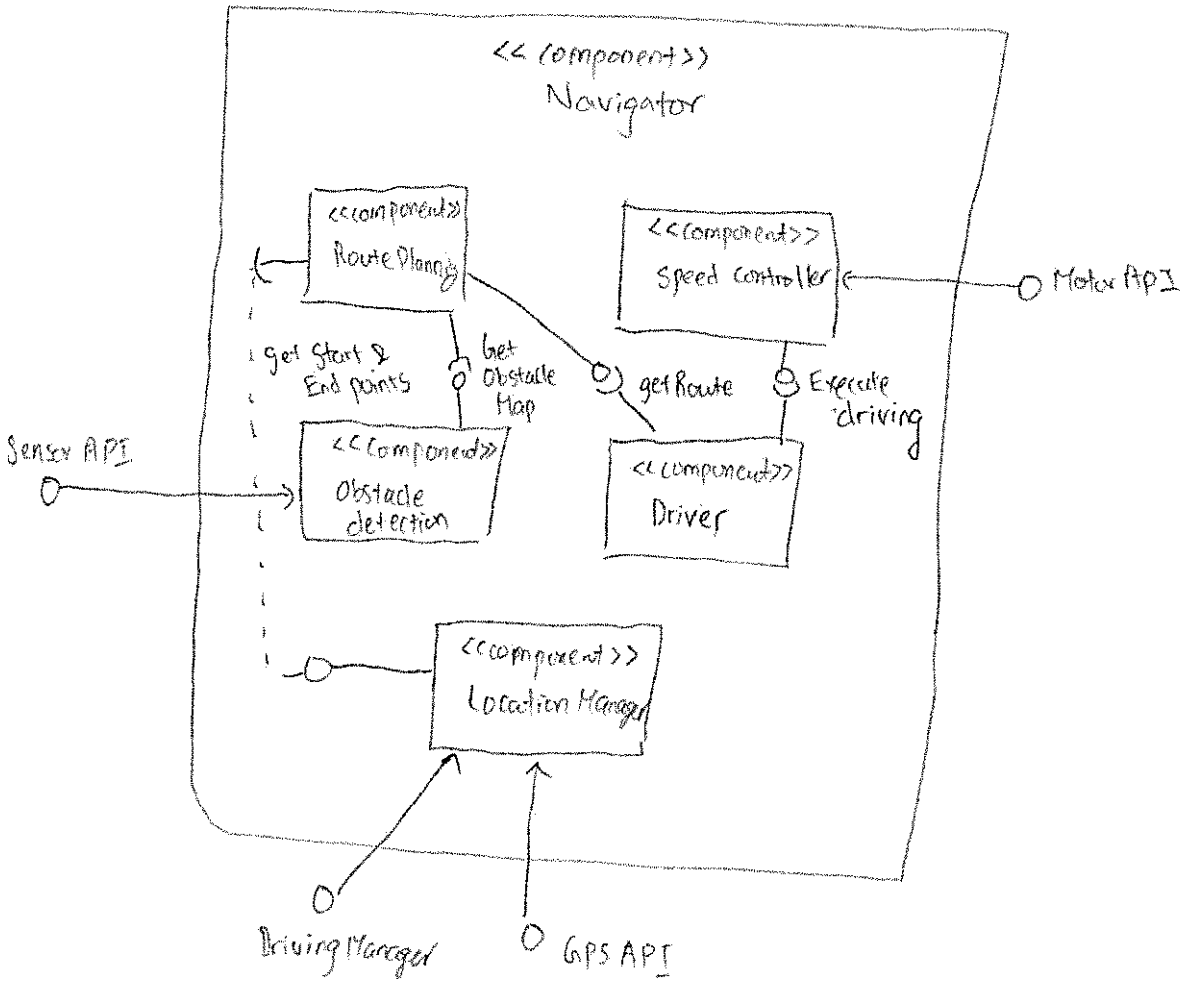
(b)



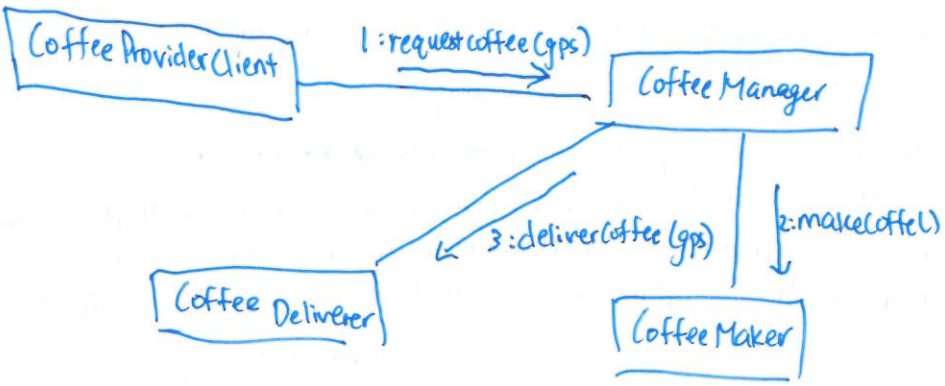


[Design is subjective; Each junior software architect might have his own design]

3) (d)


<<component>>
Navigator
<<component>>
Route Planner
<<component>>
Speed controller
Motor API
get Start & End points
Set Obstacle Map
getRoute
Execute driving
Sensor API
<<component>>
Obstacle detection
<<component>>
Driver
<<component>>
Location Manager
Driving Manager
GPS API

4) (a)



(b) Add security component to Mobile App GUI.
Add login function to the mobile app so that people can be authenticated before placing orders.
Only office staff are allowed to place orders and we verify that they are as such here.
Users have no interaction with ordering of coffee other than at this point.

(c) Health and Safety Committee. Concerns could be raised as to the usage of robots to make consumerable beverages. Robots might not be able to detect defects in coffee products the way a human can.

(d) Fascade class design pattern. 10 classes.
All components need to be modeled by one fascade classes at least. That gives 5 fascade classes.
Including the other 5 classes, this brings the tally to 10 classes.

(e) Composition of product is static while system is dynamic. Product is made up of fixed number of parts. A system may have parts constantly added or removed.
Reuse in a system is extensive while that in a product is minimal. Products are specific to a task, System provide for many similar task and employ sub groups repeatedly to achieve these task.
A system has a single instance while products have many instances. A system stands alone while a product might be used repeatedly in several different settings and times.

(f) Product certification ensues a specificity that accomodates the non trivial details of a task, this makes it incompatible with other task.