PRINCIPLES OF SOFTWARE DESIGN AND DEVELOPMENT HOMEWORK-2

İSAK ÇOBAN
OĞUZHAN ERÇELİK
SEVDA ERGÜN

PHARMACY ON DUTY

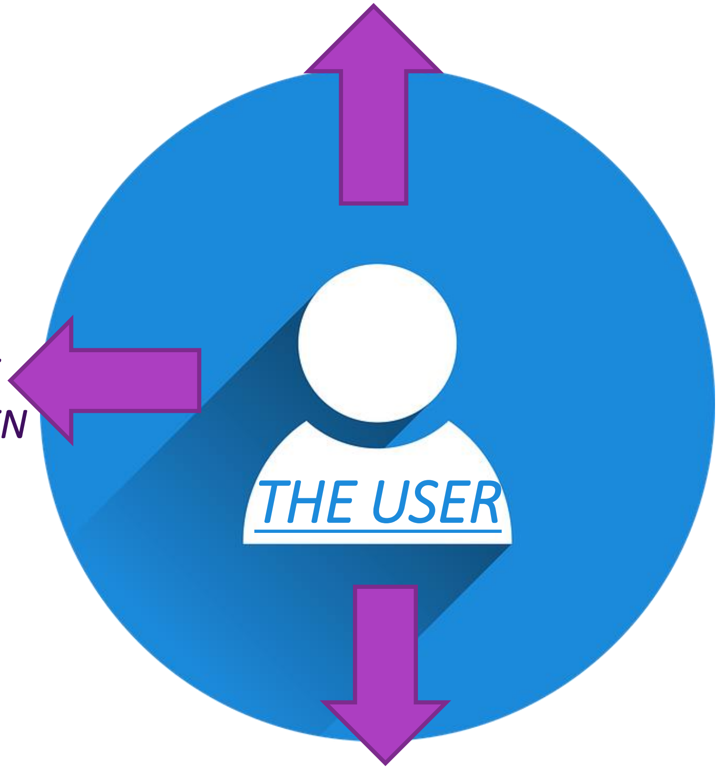
Health is a really important subject for us. Nowadays diseases are increasing in our world. There is at least one patient who has to take pills constantly in pretty much every houses. Especially in big cities finding a pharmacy can be a disaster. For instance; losing the road, not knowing the pharmacies working time etc… There are a lot of people who has this problem so there must be a solution for this. In fact there are a lot of solutions and one of them is using a traditional map. This would take a lot of time and effort. Writing a software on a mobile phone is one of the best solutions for this problem.

One of our clients came to us for the same problem. Alex lives with a patient. He travels a lot and an unexpected phone call from his home can panic him. When he doesn't know anywhere around him, he can open his GPS and find the nearest pharmacy. But sometimes he gets that call while he was driving somewhere and he wants to find and go to pharmacy without any deviation. Because of his job he plans business trips every month. He wants to be prepared in case of needing a pharmacy. He sets the date and the city-town like he wants and can see all the information about pharmacies around that city and town. Due to those wishes our client asked us to write a program.

REQUIREMENT LIST

1. A detailed world map to see every pharmacies.
2. Locations of all pharmacies to compare them with each other.
3. Information of all pharmacies like adresses, names etc to inform the user.
4. Location of the user to calculate the distance between the user and nearby pharmacies.
5. Every pharmacy owner must enter the information that whether they are on duty or not. With this information the software can know which pharmacies to select for comparison process.
6. The location that the user wants to go.
7. The date and the city if the user wants to search pharmacies in a city on a specific date.
8. A navigator shows how to go to the pharmacy.

USE CASE DIAGRAM



THE USER

FINDING NEAREST PHARMACY WITH NAVIGATON

FINDING THE BEST PHARMACY BETWEEN TWO LOCATIONS

FINDING PHARMACIES AND INFORMATION ON DUTY DUE TO DATE

USE CASE 1:
FINDING NEAREST PHARMACY WITH NAVIGATON

FLOW OF EVENTS:

1. The user needs to find nearest pharmacy and go immediately.
2. The user activates GPS.
3. The application perceives the location on the map.
4. The application scans a circle with radius of 1 km on the map.
5. Pharmacies on duty remaining in the area are determined.

   5.1 No on duty pharmacy is determined and was sent a warning to the program.

   5.2 Returns to 4 with a larger circle until there are pharmacies on duty then continue.

6. The program marks them on the map.
7. The application compares distances of pharmacies on duty.
8. The application displays the nearest pharmacy on duty.
9. The user sees on the screen how to go to the pharmacy.
10. The user goes to that location.

EXTERNAL INITIATOR → THE USER

- ***Start Condition:*** Being in need of finding and finding the nearest pharmacy.

- ***Stop Condition:*** Getting the direction.

- ***Clear Value:*** The user can find the nearest pharmacy without taking too much time and get the direction of it.

USE CASE 2:
FINDING THE BEST PHARMACY BETWEEN TWO LOCATIONS

FLOW OF EVENTS:

1. The user gets lost and wants to find nearby pharmacies and go the nearest one.
2. The user opens the program and activates GPS.
3. The user enters the location that he/she wants to go.
4. The application perceives both location on the map.
5. The application surrounds two locations on the map as one circle.
6. Pharmacies on duty remaining in the area is determined.
7. The program find the shortest way between two locations.
8. The program marks them on the map.
9. The program determines the best pharmacy on the way by comparing the distances. (The pharmacy that the user can go without any deviation from the road means the best pharmacy.)
10. The program displays the best pharmacy and its road.
11. The user goes to the pharmacy.

EXTERNAL INITIATOR → THE USER

- ***Start Condition:*** Getting lost and being in need of going a nearby pharmacy.

- ***Stop Condition:*** Displaying the best pharmacy and its road.

- ***Clear Value:*** The goal is to find a pharmacy without deviating from the road.

USE CASE 3:
FINDING PHARMACIES AND INFORMATION ON DUTY DUE TO DATE

FLOW OF EVENTS:

1. The user wants to learn pharmacies' information in a city on a date .
2. The user opens the program.
3. The user enters a date for when to search pharmacies.
4. The user enters a city and town for where to search pharmacies.
5. The application perceives the city and the date (destination).
6. The program scans the city by destination and determines on duty pharmacies on the given date.
7. The program displays pharmacies on duty and their information as a list such as their names, phone numbers, addresses etc.

EXTERNAL INITIATOR → THE USER

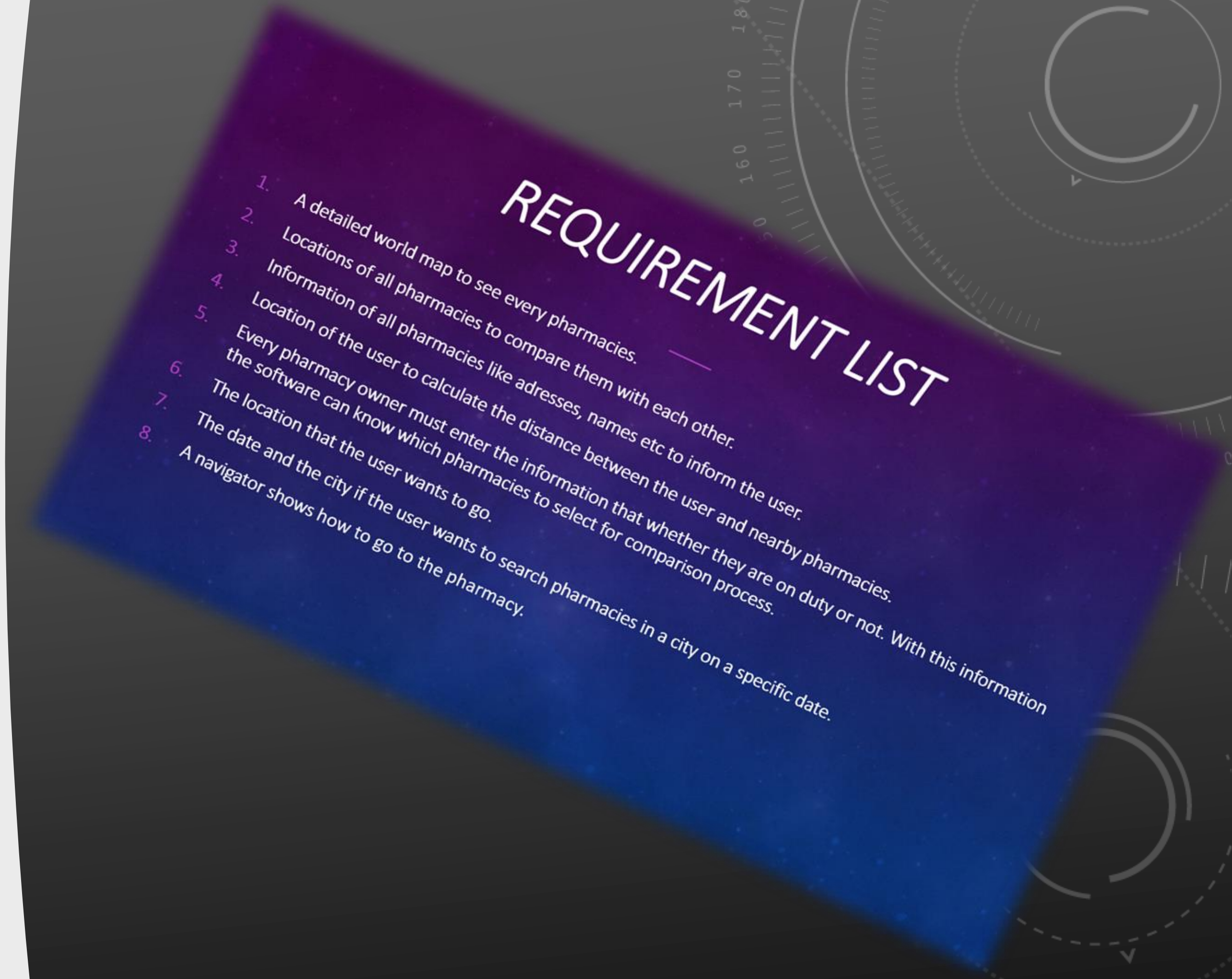
- ***Start Condition:*** Entering a date, a city.

- ***Stop Condition:*** Displaying pharmacies’ information.

- ***Clear Value:*** The user knows the pharmacies on duty and their information in a city which he/she will go in future.

MATCHING REQUIREMENT LIST WITH USE CASE STEPS

1. The user needs to find nearest pharmacy and go immediately.(N/A)
2. The user activate GPS.(4)
3. The application perceives the location on the map.(N/A)
4. The application scan a circle with radius of 1 km on the map.(1)
5. Pharmacies on duty remaining in the area is determined.(2)
   5.1 No pharmacy on duty is determined and was sent a warning to the program.
   5.2 Returns to 4 with a larger circle until there are pharmacies on duty then continue.
6. The program marks them on the map.(5)
7. The application compares distances of pharmacies on duty.(2)
8. The application displays the nearest pharmacy on duty. (N/A)
9. The user sees on the screen how to go to the pharmacy.(8)
10. The user goes to that location. (N/A)

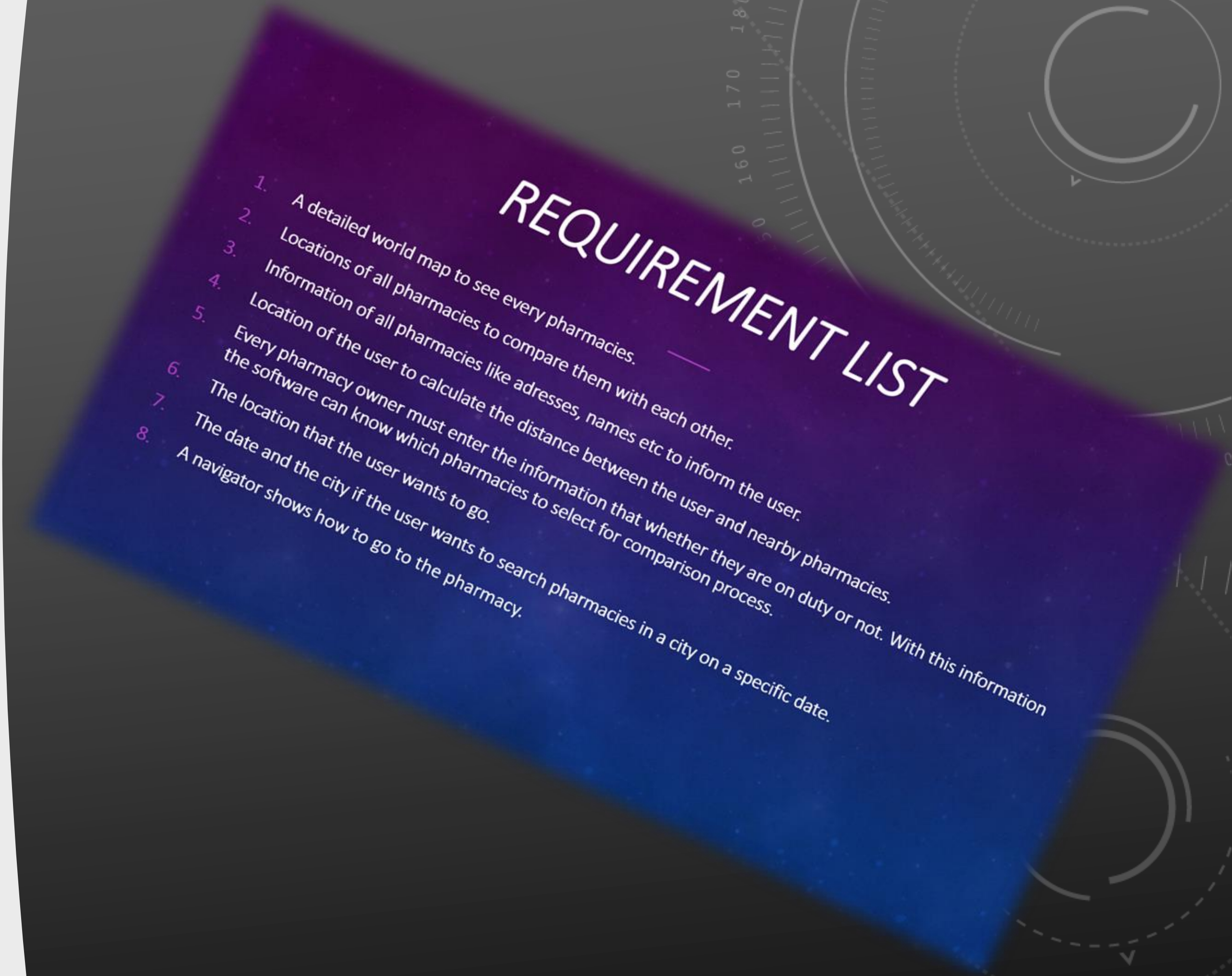


REQUIREMENT LIST

1. A detailed world map to see every pharmacies.
2. Locations of all pharmacies to compare them with each other.
3. Information of all pharmacies like adresses, names etc to inform the user.
4. Location of the user to calculate the distance between the user and nearby pharmacies.
5. Every pharmacy owner must enter the information that whether they are on duty or not. With this information the software can know which pharmacies to select for comparison process.
6. The location that the user wants to go.
7. The date and the city if the user wants to search pharmacies in a city on a specific date.
8. A navigator shows how to go to the pharmacy.

MATCHING REQUIREMENT LIST WITH USE CASE STEPS

1. The user gets lost and wants to find nearby pharmacies and go the nearest one. (N/A)
2. The user opens the program and activates GPS.(4)
3. The user enters the location that he/she wants to go.(6)
4. The application perceives both location on the map. (N/A)
5. The application surrounds two locations on the map as one circle. (N/A)
6. Pharmacies on duty remaining in the area is determined.(2)
7. The program find the shortest way between two locations.(5)
8. The program marks them on the map.(4)
9. The program determines the best pharmacy on the way by comparing the distances. (The pharmacy that the user can go without any deviation from the road means the best pharmacy.) (N/A)
10. The program displays the best pharmacy and its road.(8)
11. The user goes to the pharmacy. (N/A)

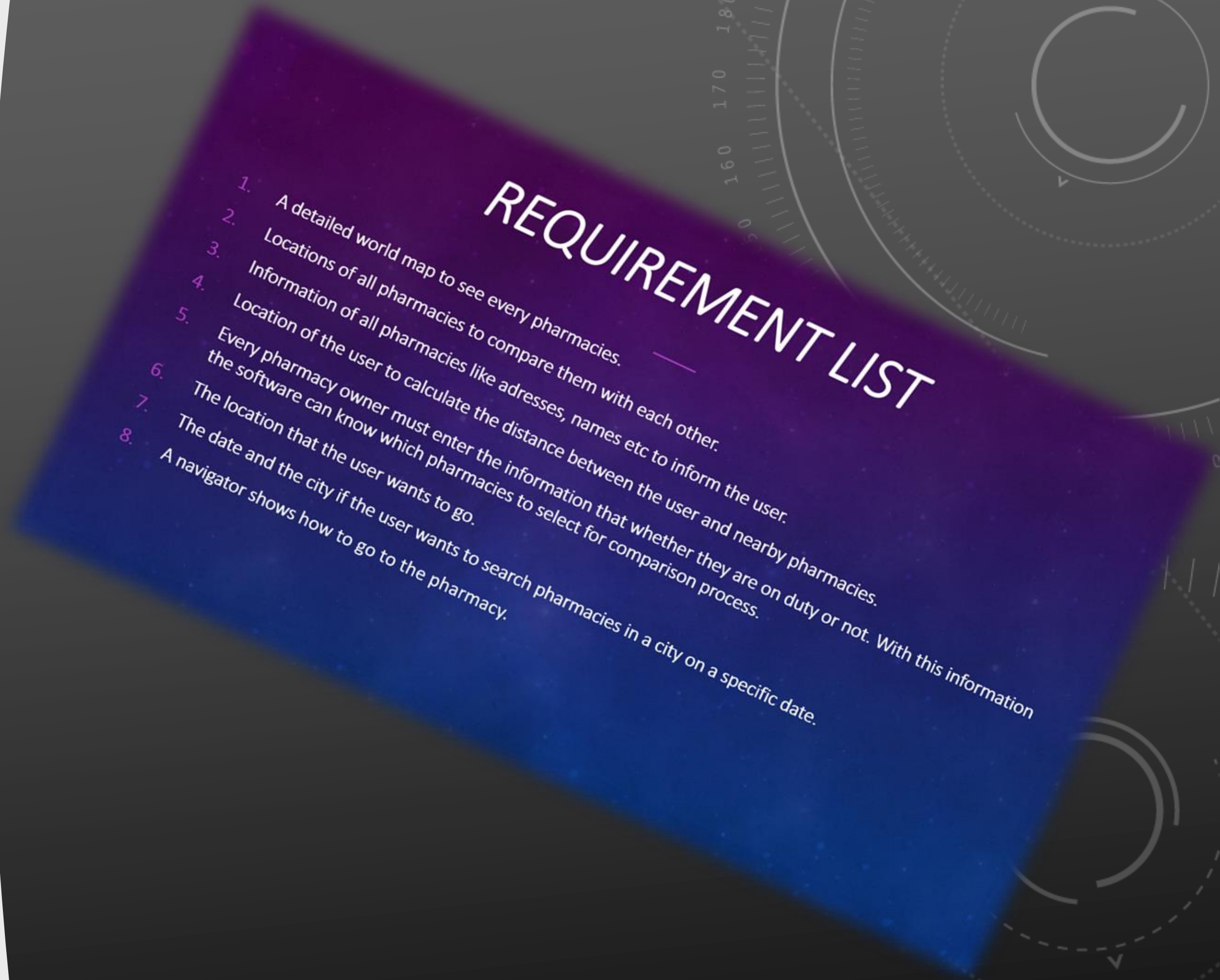


REQUIREMENT LIST

1. A detailed world map to see every pharmacies.
2. Locations of all pharmacies to compare them with each other.
3. Information of all pharmacies like adresses, names etc to inform the user.
4. Location of the user to calculate the distance between the user and nearby pharmacies.
5. Every pharmacy owner must enter the information that whether they are on duty or not. With this information the software can know which pharmacies to select for comparison process.
6. The location that the user wants to go.
7. The date and the city if the user wants to search pharmacies in a city on a specific date.
8. A navigator shows how to go to the pharmacy.

MATCHING REQUIREMENT LIST WITH USE CASE STEPS

1. The user wants to learn pharmacies' information in a city on a date. (N/A)
2. The user opens the program. (N/A)
3. The user enters a date for when to search pharmacies.(7)
4. The user enters a city and town for where to search pharmacies.(7)
5. The application perceives the city and the date (destination). (N/A)
6. The program checks on the map if pharmacies in the city are on duty or not on that day.(5)
7. The program displays pharmacies on duty and their information as a list such as their names, phone numbers, addresses etc.(5)



REQUIREMENT LIST

1. A detailed world map to see every pharmacies.
2. Locations of all pharmacies to compare them with each other.
3. Information of all pharmacies like adresses, names etc to inform the user.
4. Location of the user to calculate the distance between the user and nearby pharmacies.
5. Every pharmacy owner must enter the information that whether they are on duty or not. With this information the software can know which pharmacies to select for comparison process.
6. The location that the user wants to go.
7. The date and the city if the user wants to search pharmacies in a city on a specific date.
8. A navigator shows how to go to the pharmacy.

TEXTUAL ANALYSIS OF FIRST USE CASE

1. The user needs to find nearest pharmacy and go immediately.
2. The user activates GPS.
3. The application perceives the location on the map.
4. The application scans a circle with radius of 1 km on the map.
5. Pharmacies on duty remaining in the area is determined.

   5.1 No pharmacy on duty is determined and was sent a warning to the program.

   5.2 Returns to 4 with a larger circle until there are pharmacies then continue.
6. The program marks them on the map.
7. The application compares distances of pharmacies on duty.
8. The application displays the nearest pharmacy on duty.
9. The user sees on the screen how to go to the pharmacy.
10. The user goes to that location.

🔵 *Verbs that are candidate to be method*

🟢 *Nouns that are candidate to be class*

TEXTUAL ANALYSIS OF SECOND USE CASE

1. The user gets lost and wants to find nearby pharmacies and go the nearest one.
2. The user opens the program and activates GPS.
3. The user enters the location that he/she wants to go.
4. The application perceives both location on the map.
5. The application surrounds two locations on the map as one circle on the map.
6. Pharmacies on duty remaining in the area is determined.
7. The program finds the shortest way between two locations.
8. The program marks on the map.
9. The program determines the best pharmacy on the way by comparing the distances. (The pharmacy that the user can go without any deviation from the road means the best pharmacy.)
10. The program displays the best pharmacy and its road.
11. The user goes to the pharmacy.

🔵 *Verbs that are candidate to be method*

🟢 *Nouns that are candidate to be class*

TEXTUAL ANALYSIS OF THIRD USE CASE

1. The user wants to learn pharmacies’ information in a city on a date .
2. The user opens the program.
3. The user enters a date for when to search pharmacies.
4. The user enters a city and town for where to search pharmacies.
5. The application perceives the city and the date (destination).
6. The program checks on the map if pharmacies in the city are on duty or not on that day.
7. The program displays pharmacies on duty and their information as a list such as their names, phone numbers, addresses etc.

🔵 *Verbs that are candidate to be method*

🟢 *Nouns that are candidate to be class*

TEXTUAL ANALYSIS OF USE CASES

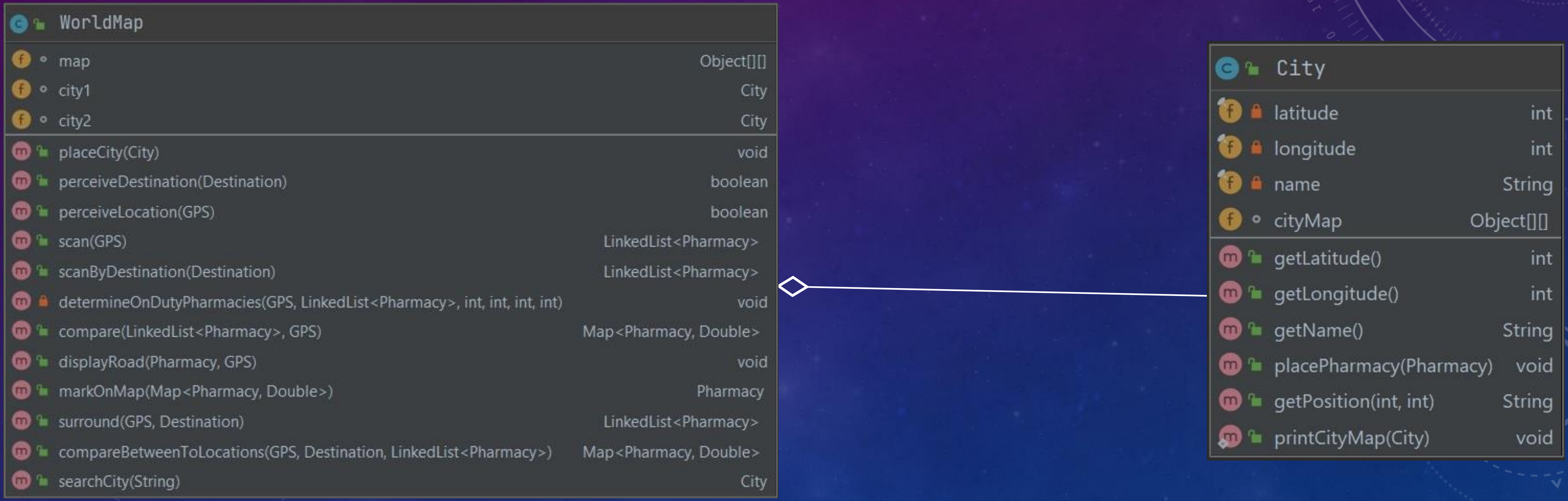
Methods

- Find
- Perceive
- Determine
- Scan
- Display
- Compare
- Surround
- Search
- Mark

Classes

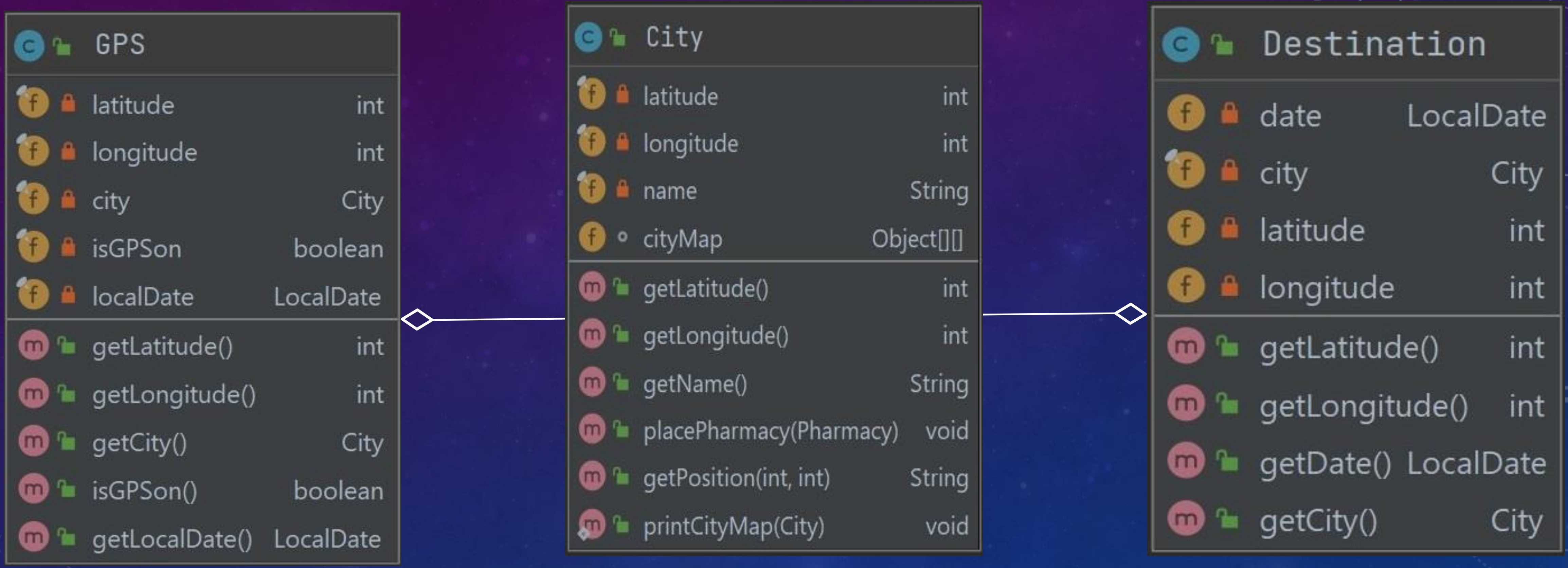
- Pharmacy
- GPS
- WorldMap
- Test
- Destination
- City

UML DIAGRAM



WorldMap	
map	Object[][]
city1	City
city2	City
placeCity(City)	void
perceiveDestination(Destination)	boolean
perceiveLocation(GPS)	boolean
scan(GPS)	LinkedList<Pharmacy>
scanByDestination(Destination)	LinkedList<Pharmacy>
determineOnDutyPharmacies(GPS, LinkedList<Pharmacy>, int, int, int, int)	void
compare(LinkedList<Pharmacy>, GPS)	Map<Pharmacy, Double>
displayRoad(Pharmacy, GPS)	void
markOnMap(Map<Pharmacy, Double>)	Pharmacy
surround(GPS, Destination)	LinkedList<Pharmacy>
compareBetweenToLocations(GPS, Destination, LinkedList<Pharmacy>)	Map<Pharmacy, Double>
searchCity(String)	City

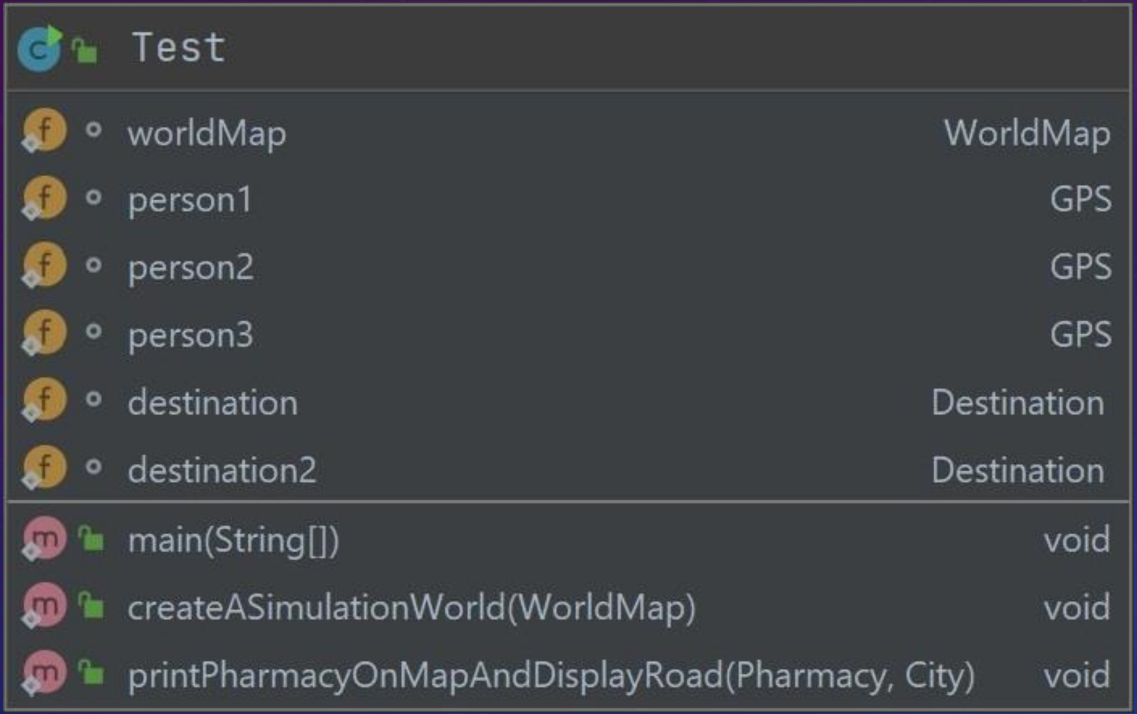
City	
latitude	int
longitude	int
name	String
cityMap	Object[][]
getLatitude()	int
getLongitude()	int
getName()	String
placePharmacy(Pharmacy)	void
getPosition(int, int)	String
printCityMap(City)	void



GPS	
latitude	int
longitude	int
city	City
isGPSon	boolean
localDate	LocalDate
getLatitude()	int
getLongitude()	int
getCity()	City
isGPSon()	boolean
getLocalDate()	LocalDate

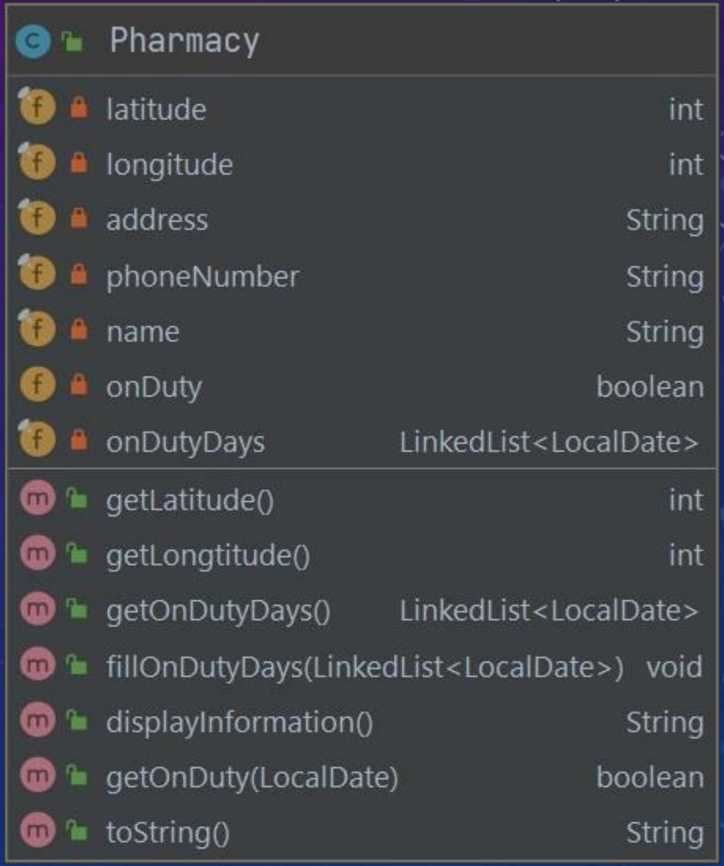
City	
latitude	int
longitude	int
name	String
cityMap	Object[][]
getLatitude()	int
getLongitude()	int
getName()	String
placePharmacy(Pharmacy)	void
getPosition(int, int)	String
printCityMap(City)	void

Destination	
date	LocalDate
city	City
latitude	int
longitude	int
getLatitude()	int
getLongitude()	int
getDate()	LocalDate
getCity()	City



Test

worldMap	WorldMap
person1	GPS
person2	GPS
person3	GPS
destination	Destination
destination2	Destination
main(String[])	void
createASimulationWorld(WorldMap)	void
printPharmacyOnMapAndDisplayRoad(Pharmacy, City)	void



Pharmacy

latitude	int
longitude	int
address	String
phoneNumber	String
name	String
onDuty	boolean
onDutyDays	LinkedList<LocalDate>
getLatitude()	int
getLongtitude()	int
getOnDutyDays()	LinkedList<LocalDate>
fillOnDutyDays(LinkedList<LocalDate>)	void
displayInformation()	String
getOnDuty(LocalDate)	boolean
toString()	String