POLITECNICO DI MILANO

MyTaxiService

Authors: Alessandro Comodi
Davide Conficconi
Stefano Longari

Index

MyTaxiService Project:

- Requirements Analysis & Specification
- Design
- Integration Test Plan
- Project Plan

Code Inspection

Requirement Analysis & Specifications

Functionalities of the final product:

- Guest:
  - Register and login
- User:
  - Request or reserve a taxi w/ or w/o sharing option
  - Manage taxi requests and reservations
  - Manage profile
- Taxi Driver:
  - Manage availability
  - Confirm or decline requests

Requirement Analysis & Specifications

User characteristics:

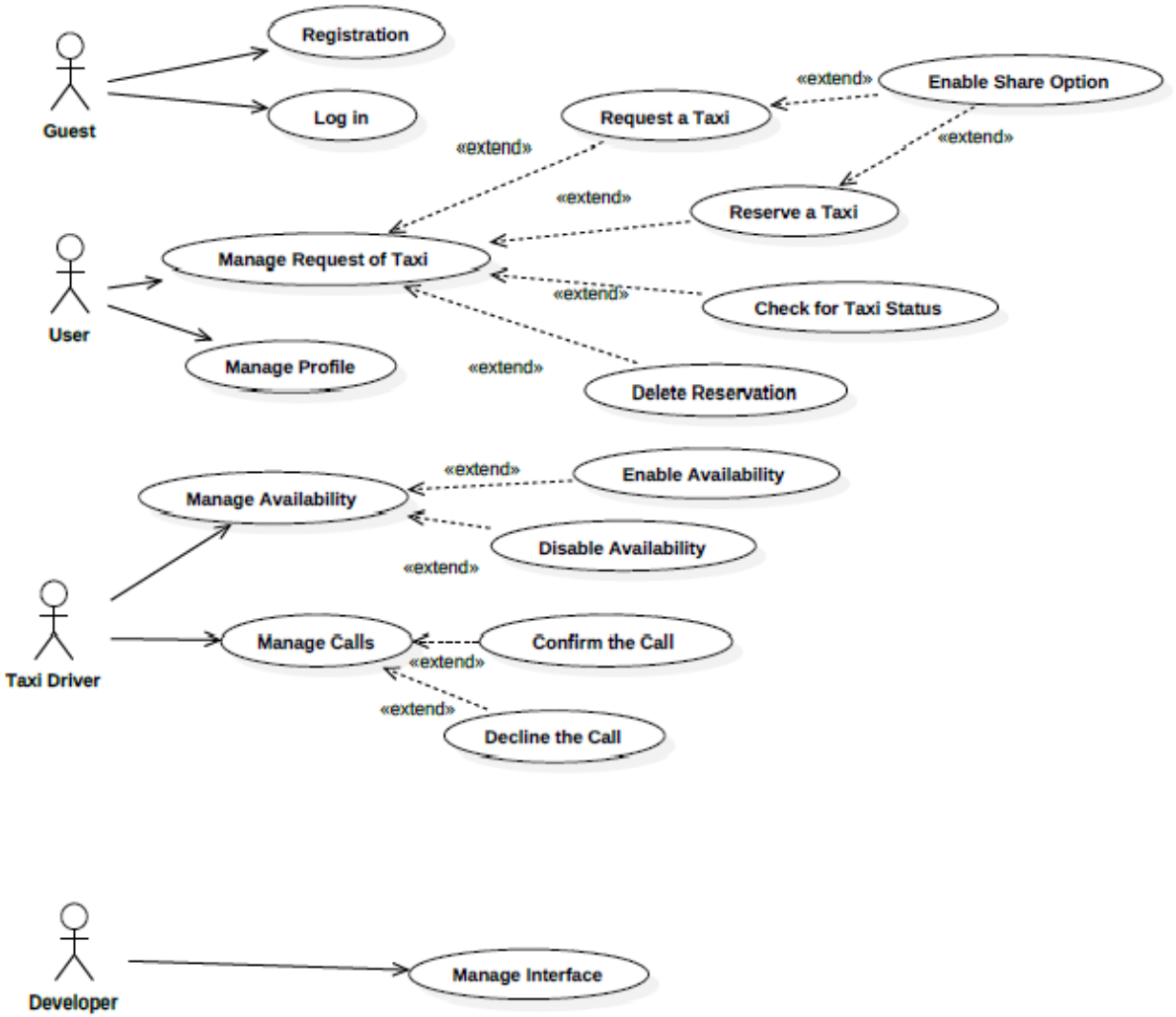
- Basic web knowledge
- Internet enabled devices
  - (taxi drivers only) GPS service on smartphone

General assumptions

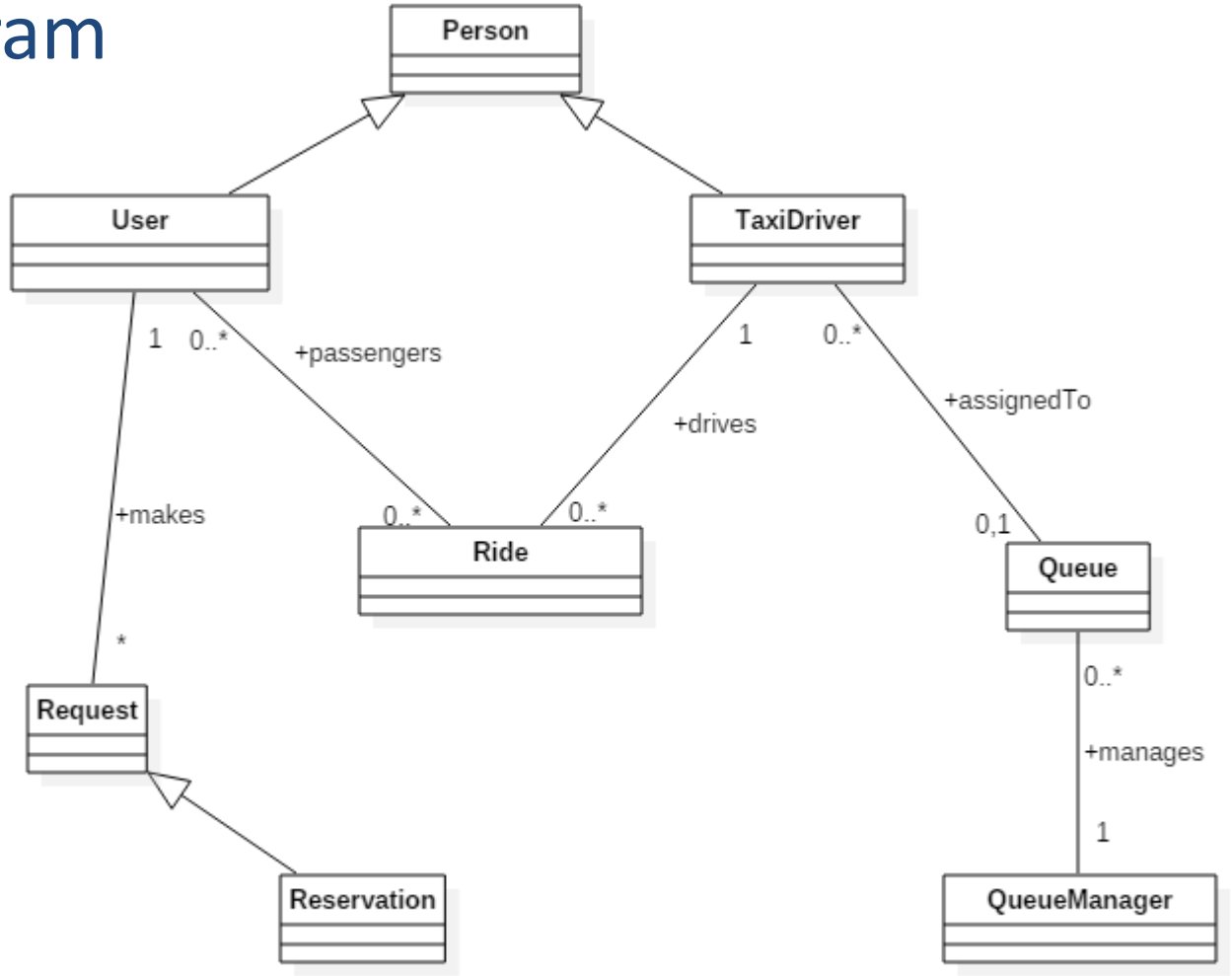
- Cover most of the mobile operating systems (iOS, Android, ect.)
- The target city population is around 2-3 millions
- Taxi-Driver side application is not available on marketplaces.

Requirement Analysis & Specifications

Use Case Diagram

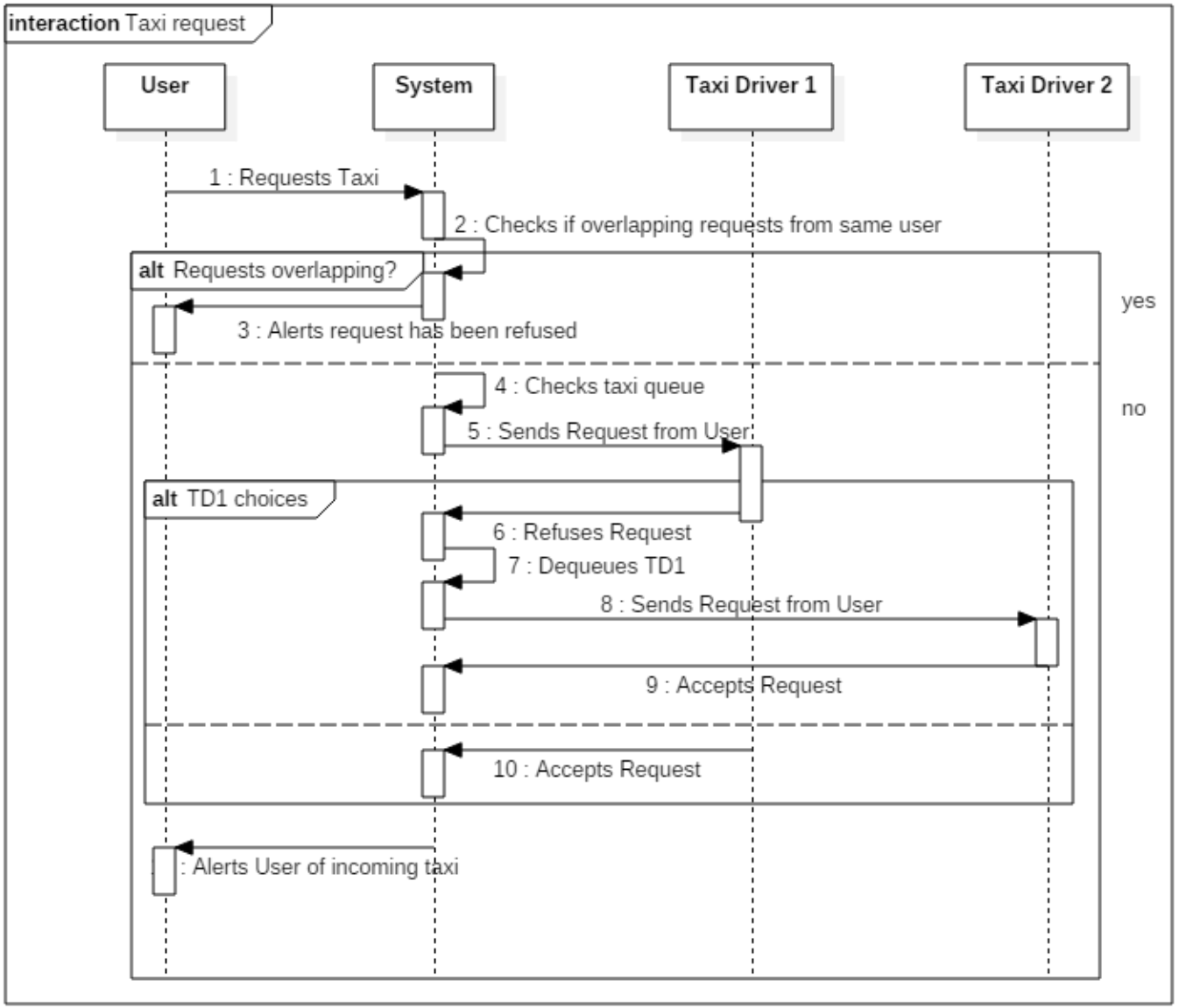


Class Diagram



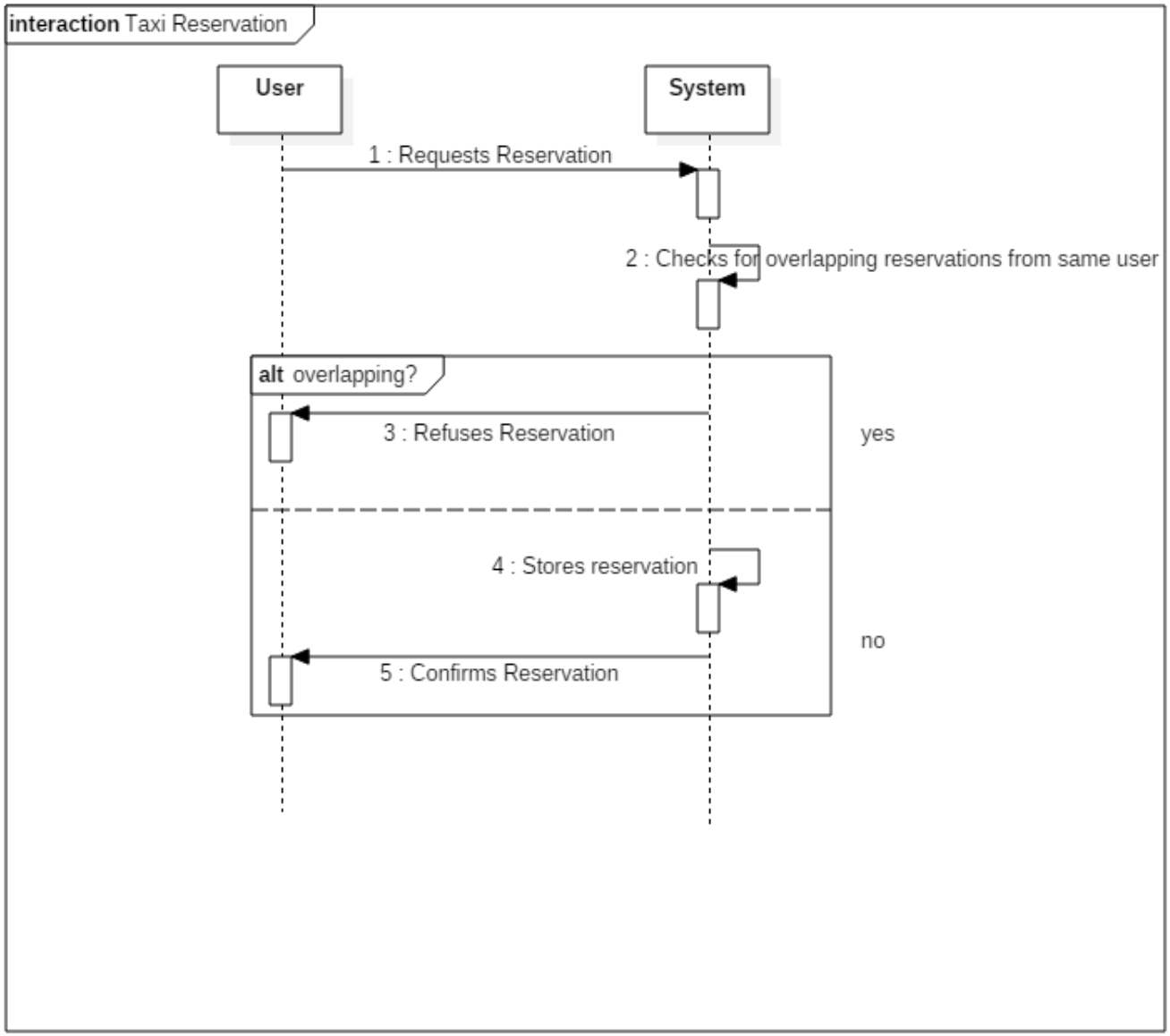
Requirement Analysis & Specifications

Sequence Diagram (1)



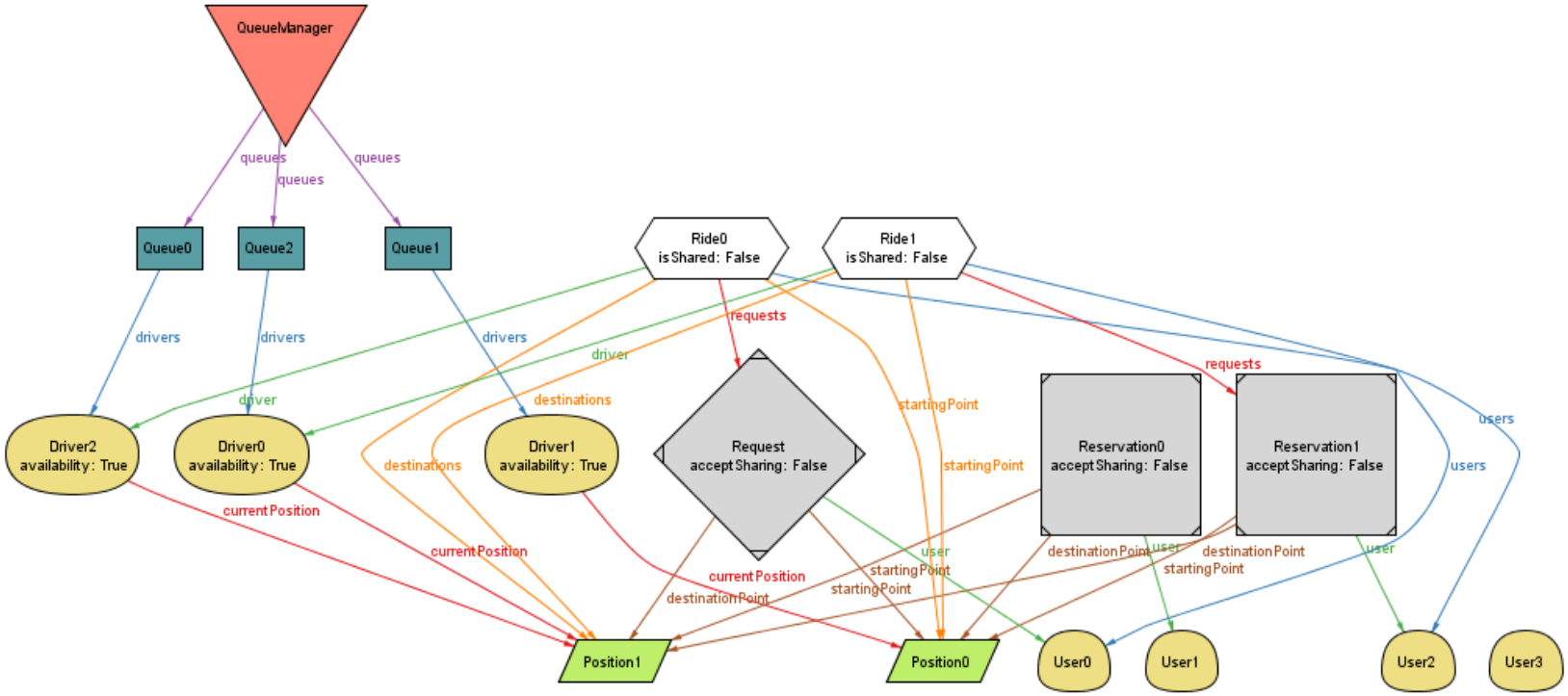
Requirement Analysis & Specifications

Sequence Diagram (2)



Requirement Analysis & Specifications

Alloy World



Design

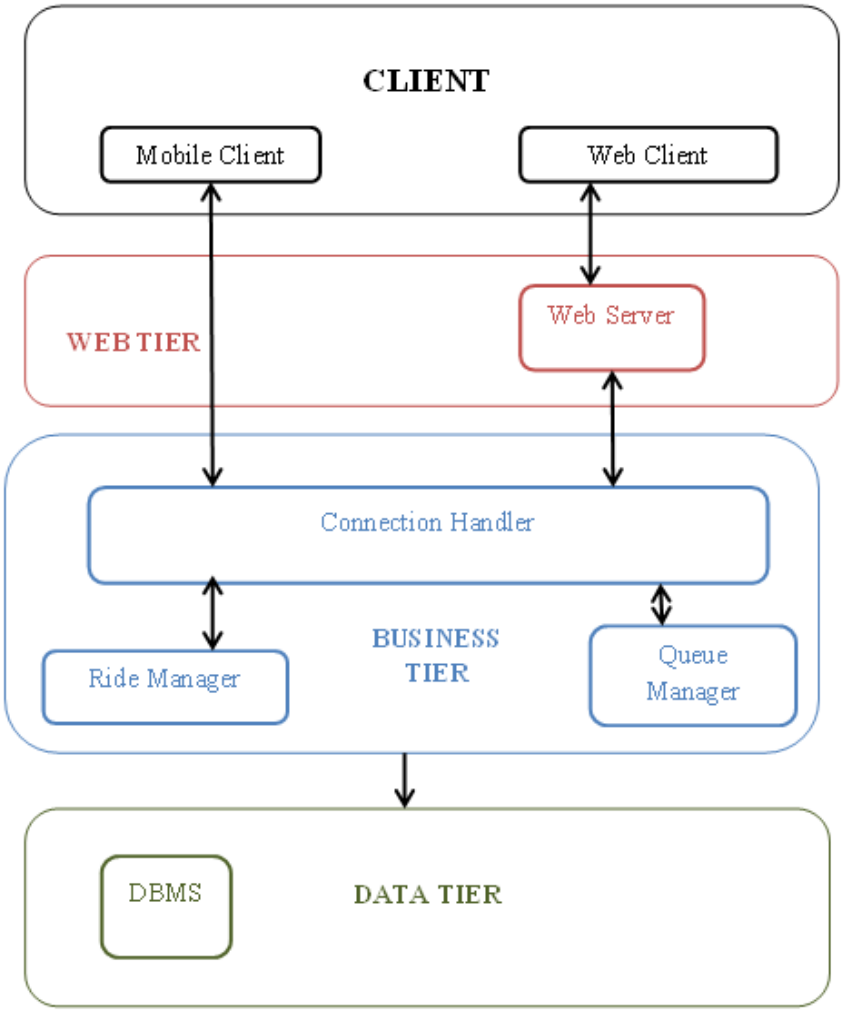
The design section deals with the technical point of view of the project:

- Overall architecture
- Relevant Components
- Main Algorithms
- Mockups

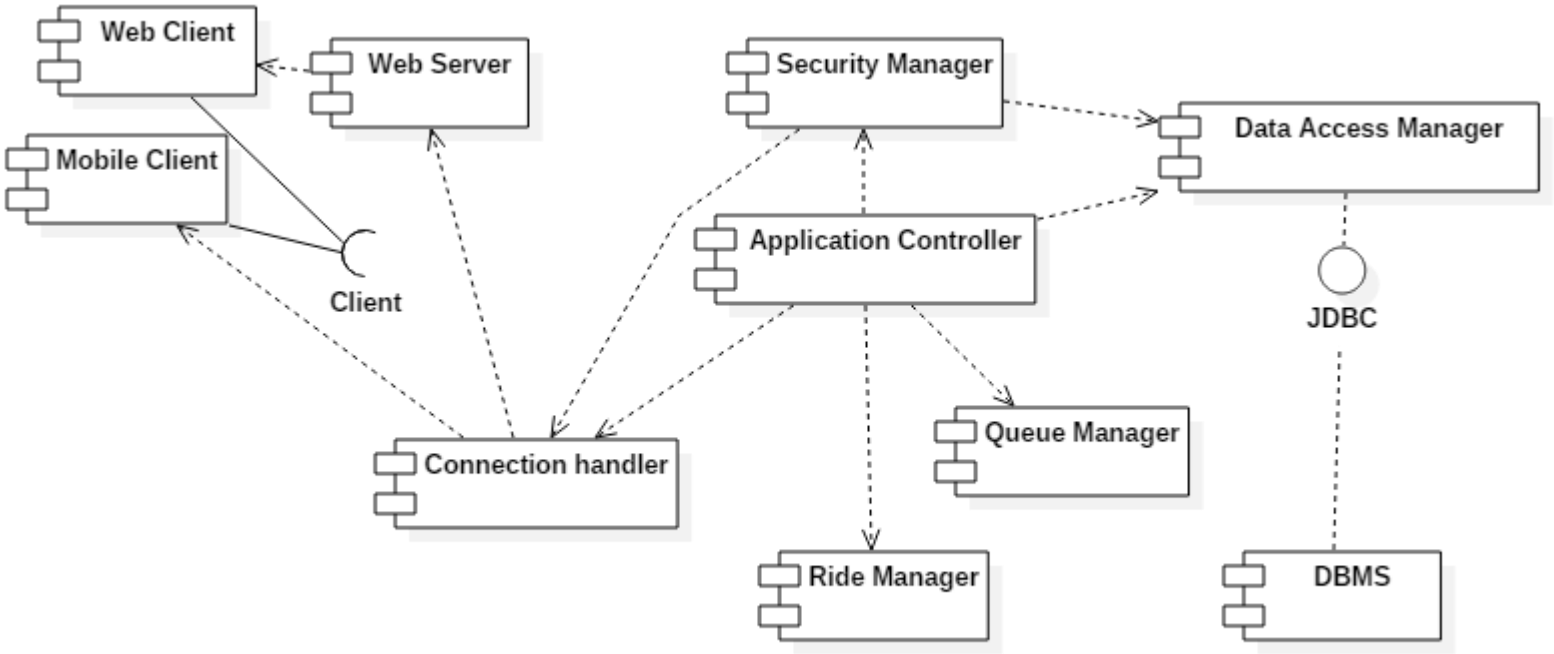
Design

Software Architecture:

- Multi-tier



Component view



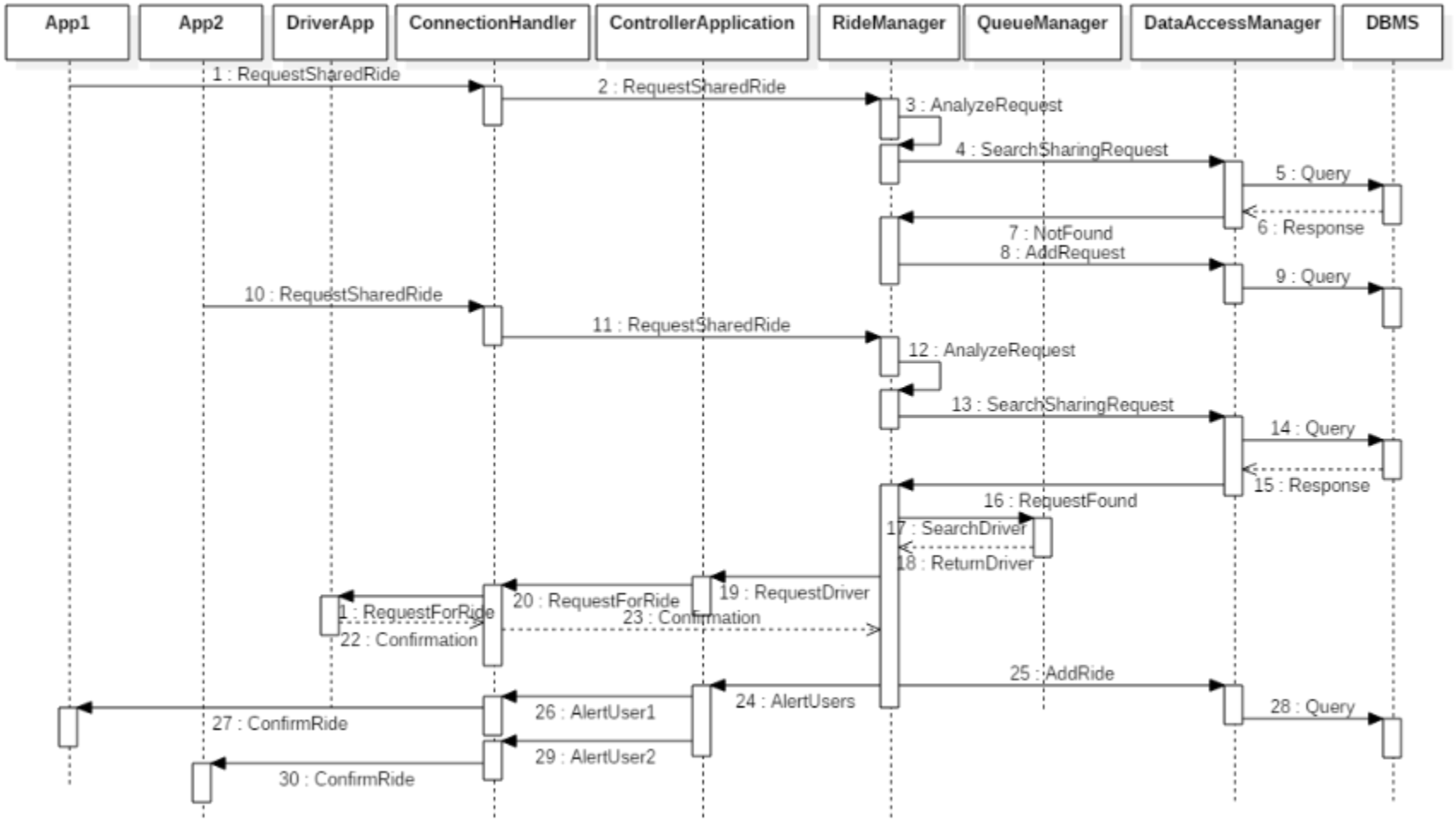
Design

Relevant Components:

- Mobile/Web client
- Connection Handler
- Application Controller
- Queue Manager

Design

Shared ride sequence diagram



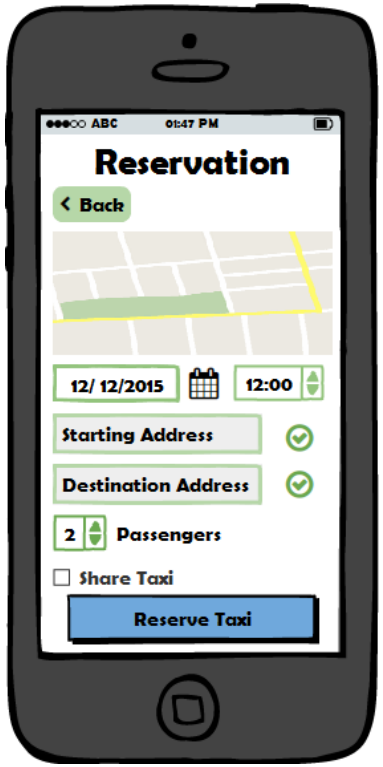
Description of the Queue Manager

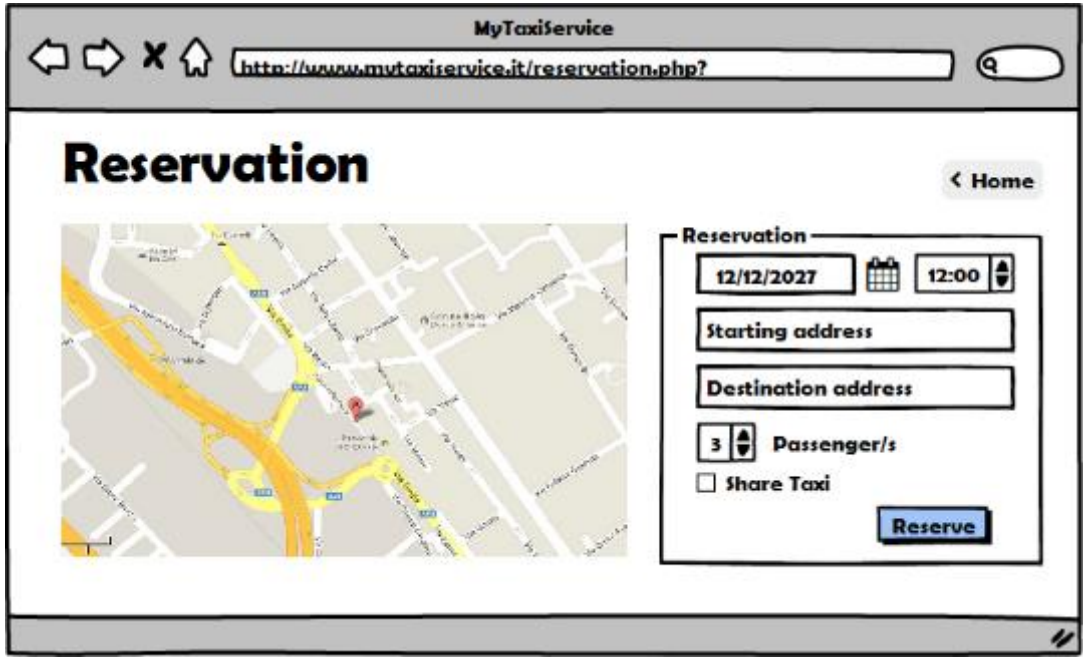
Algorithms:

- Positioning
  - To distribute taxis to avoid overpopulation/lack of drivers
  - Check drivers GPS/requests per hour in zones
  - Eventually redirects drivers
- Distribution prevision
  - Register rides and analyze distribution

Design

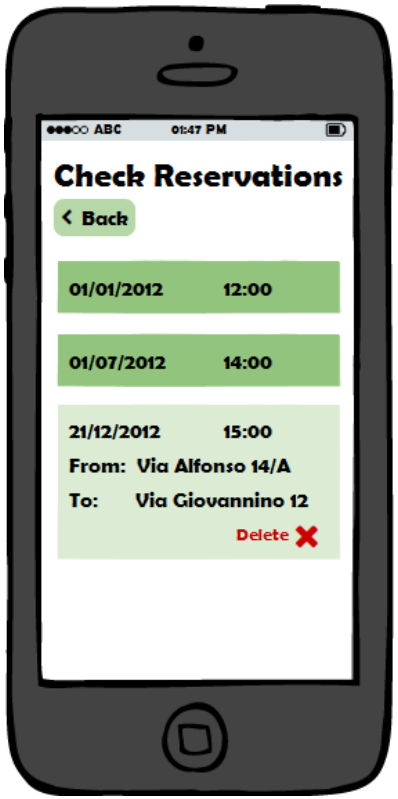
Mockups

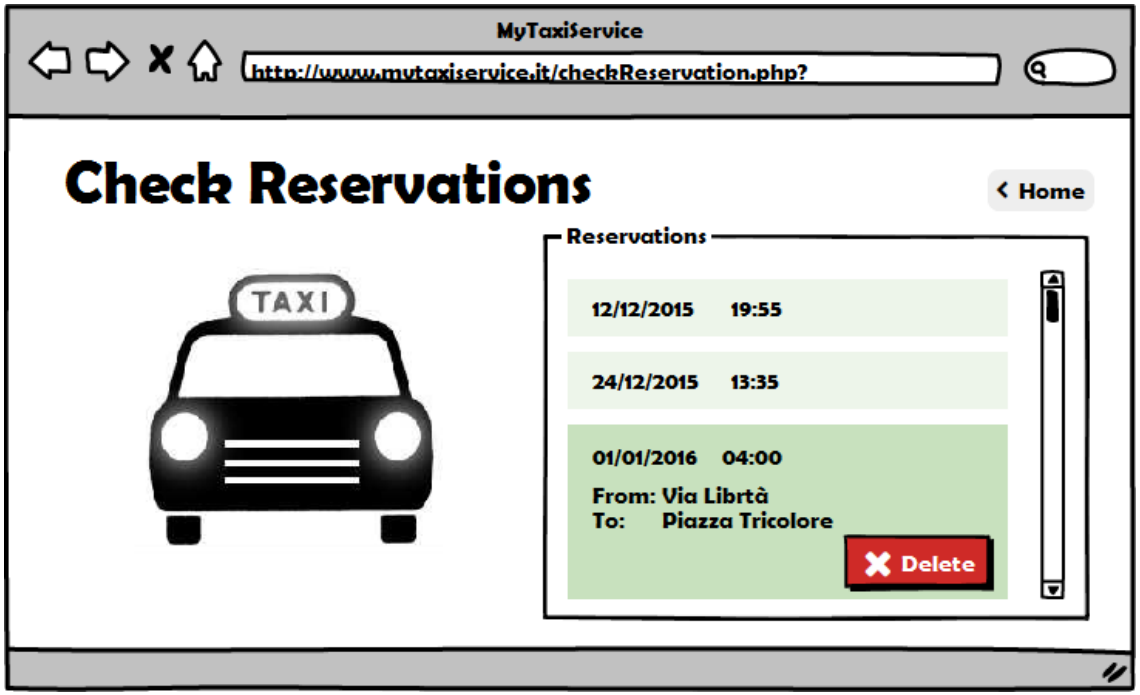




Design

Mockups





Integration Test Plan

Entry criteria:

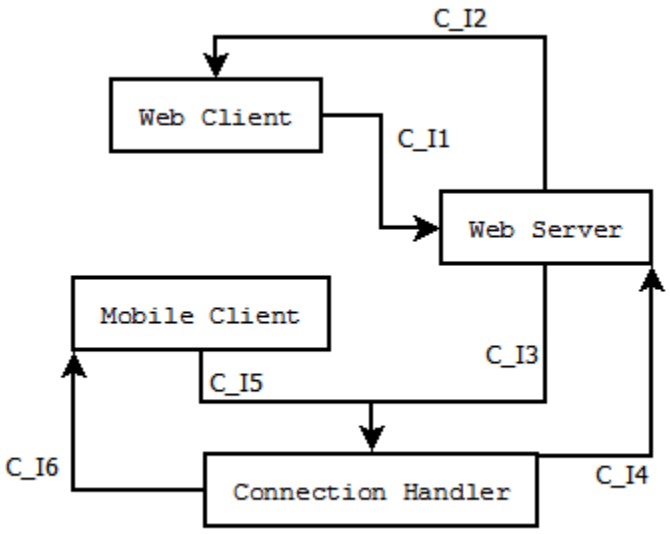
- Unit tests
- Fulfilled requirements of RASD and DD

Integration Strategy:

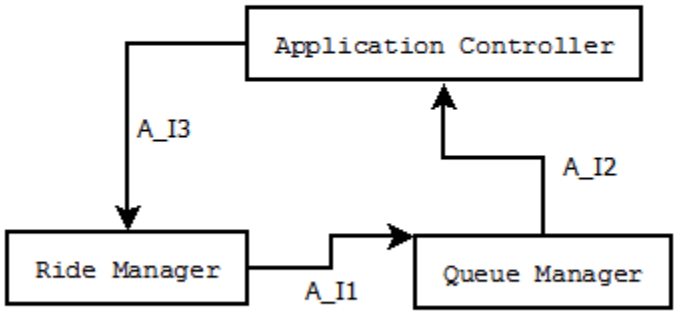
- Bottom Up
- Functional grouping

Integration Test Plan

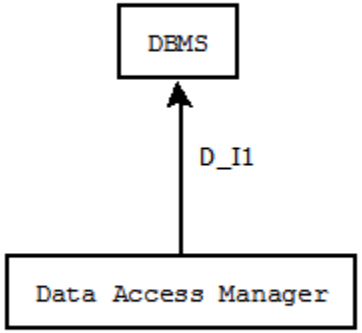
Module integration sequence



Communication Subsystem



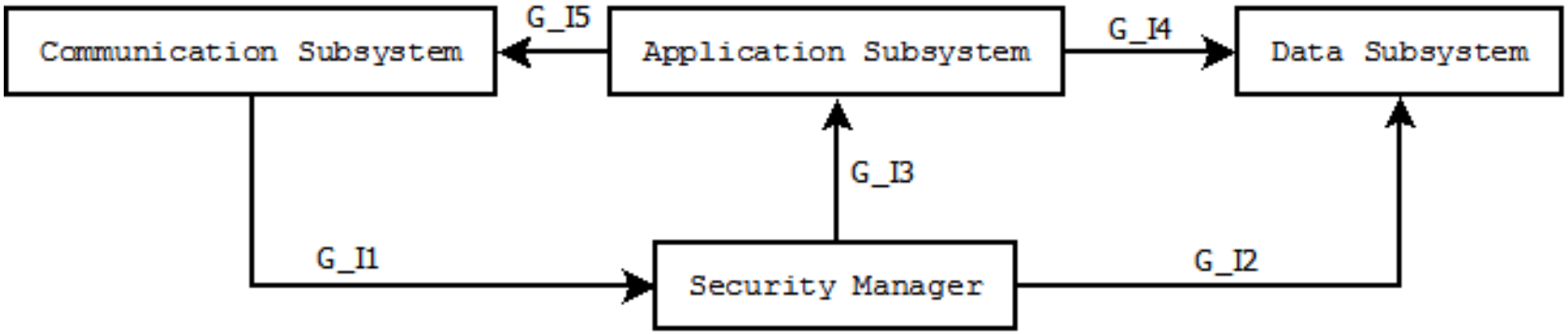
Application Subsystem



Data Subsystem

Integration Test Plan

Sybsystem integration sequence



Integration Test Plan

Individual Step example

3.4.6 Application Subsystem → Communication Subsystem

Test case identifier	G_I5T1
Components involved	Application Subsystem → Communication Subsystem
Input specifications	Notification from the Application subsystem directed to a client
Purposes	Check if the Communication Subsystem receives the right informations from the Application Subsystem
Output specifications	Notification received correctly
Environment needed	Application Subsystem and Communication Subsystem tests succeeded

Project Plan

Project size estimation:

- Attribution of function points

Function Types	Weight		
	Simple	Medium	Complex
Internal Logic Files	7	10	15
External Interface Files	5	7	10
External Inputs	3	4	6
External Inquiries	3	4	6
External Outputs	4	5	7

Project Plan

Result of size estimation

- SLOC = FP*CR = 5671
- CR = 53

Function Type	Value
Internal Logic Files	29
External Logic Files	15
External Inputs	50
External Inquiries	13
External Outputs	-
Total	107

COCOMO II:
Project effort and cost estimation

Results

Software Development (Elaboration and Construction)

Effort = 18.5 Person-months
Schedule = 9.6 Months
Cost = $33312

Total Equivalent Size = 5700 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.1	1.2	0.9	$1999
Elaboration	4.4	3.6	1.2	$7995
Construction	14.1	6.0	2.3	$25317
Transition	2.2	1.2	1.8	$3997

Project Plan

Project schedule

Month	Tasks	Developer
February		
	Queue Manager	Alessandro, Davide, Stefano
	Ride Manager	Alessandro
	Web Client	Davide
	Mobile Client	Davide
	DBMS	Stefano
March		
	Application Controller Driver	Alessandro
	Lower level components testing	Alessandro, Davide, Stefano
	Application Controller	Alessandro
	Data Access Manager	Stefano
	Web Server & Connection Handler	Davide
April		
	Application Controller	Alessandro
	Data Access Manager	Stefano
	Web Server & Connection Handler	Davide
May		
	Security Manager	Stefano
	Subsystem Linking	Alessandro, Davide
June		
	Testing Subsystems	Alessandro, Davide, Stefano
	Integration of all subsystems	Alessandro, Davide, Stefano
	Deployment	Alessandro, Davide, Stefano

Project Plan

Project risks

- Change in requirements
- No proper subject training

Code Inspection

Assigned Class:

WebDavServlet.java

Assigned Methods:

- copyResource
- doUnlock
- isLocked

Code Inspection

Relevant Issues:

25. *The class or interface declarations shall be in the following order:*

- *class/interface documentation comment:* No issues found.
- *class or interface statement:* No issues found.
- *class/interface implementation comment, if necessary:* No issues found.
- *class (static) variables (in the order: public class variables/ protected*
- *class variables/ package level/ private class variables):*
  - In class WebdavServlet this point is not respected. At line 239, 248, 262 and 277 there are protected static variables that should come before the private ones.
  - In private class WebdavStatus this point is not respected. At line 2889 we can find a private static variable followed at line 2898 by the public static variables.
- *instance variables (in the order: public instance variables/ protected instance variables/ package level/ private instance variables):* In private class LockInfo the 25.e/f points are not respected. At line 2731 we find the constructor, after which we find at line 2736 Instance variables.
- *constructors:* see above.
- *methods: No issues found.*

Code Inspection

Relevant issues:

27. *Check that the code is free of duplicates, long methods, big classes, breaking encapsulation, as well as if coupling and cohesion are adequate:*

- From what we can see WebdavServlet class is actually pretty big, with almost 3000 lines of code.
- At line 475 there is a method, doPropfind, that is about 250 lines.
- At line 923 there is a method, doLock, that is about 500 lines.
- At line 2080 there is a method, parseProperties, that is about 300 lines and has a giant elseif structure.
- At line 2366 there is a method, parseLockNullProperties, that is about 250 lines.

End