# CITS4401 Requirements and Design
# Group Project 2, Semester 1, 2020

Version: 0.2 2020-05-18
Changes: Due date; allow separate diagram submission
Please check the CITS4401 unit web page to ensure that you have the latest version.

## Project overview

You are part of a project team at the University of Western Angria (motto: "Seek sagacity"), assigned the task of developing a computer system to improve the parking situation at the University.

The University has over 20,000 students and staff, but far fewer parking bays for their cars. For a university, social interaction and being on site is an important part of research and learning activity. Students and staff wish to be able to park quickly and at reasonable cost when they need to come to the University. The University also wants to reduce car use and encourage behaviours such as public transport and car pooling. All want a fair and efficient system that supports students and staff to do their work on the University campus.

Your group will be required to:

- perform subsystem decomposition on, and review for improvements, a suggested class design diagram
- research and report on a strategy for handing persistent data

## Project rules

*Project groups*

- Project work will be done in the same groups as for Part 1 of the project.
  See the provided file `CITS4401ProjectGroups.csv` for a class list with contact details for your group members.
- All group members are required to contribute equal effort to the project. Contact Arran Stewart as soon as possible (well before the deadline) if you believe this is not the case.

*Project Deliverables and Submission Dates*

- The **final submission deadline** for your project report 4pm on Friday 22 May (week 11).

*Assessment details and guidelines*

The design project is the second part of the CITS4401 group project worth 30%. Part 2 (design) is worth 15% of the marks for CITS4401. Each group member is expected to spend around 20 hours on part 2 of the project, including background reading. Your final project must be submitted via `cssubmit` before the submission deadline. The penalties for late submission are described in the University's guidelines on assessment.

You are expected to have read and understood the University's guidelines on academic conduct. In accordance with this policy, you may discuss with students in other groups the general principles required to understand this project, but the work you submit must be the result of your own group's effort.

*Project clarification*

Please take time to read this project description carefully. Post any requests for clarification about the requirements of this assessment to help4401 so that all students may remain equally informed. Further information may be provided as required during the project. All announcements about the project will be posted on help4401.

## Project tasks

1. Review the scenario from part 1 of the project (reproduced here for your convenience).

**The Sens-O-Park System**

**The Problem**

Available parking can be difficult to find on the University campus. The Sens-O-Park System aims to solve this problem by allowing users with a mobile device to access the system and find the location of a close and available parking space.

**Goals**

The goals for the system are as follows:

- To set up and program cameras in parking areas, which will allow the system to determine whether a particular parking space is occupied or not.
- To provide secure access to the system to users, so that they can determine whether there is available parking, where it is, and how much it will cost.

**Users**

The intended users of the system are University staff, students and visitors to the University campus. They would need to have a mobile device in order to use the system.

**Mandatory features**

- The system must have cameras which can be used to determine whether a parking space is occupied or not.
- The system must be able to function in a wide range of environmental conditions – day, night, rain, etc. Angria is noted for its sometimes violent snowstorms (in winter)

and duststorms (in summer).
- System administrators should be able to add or remove parking spaces from the system or reserve them for particular staff.
- System administrators should be able to add or remove users from the system.
- The system should only be accessible to properly authenticated users.
- User details should be stored securely.
- Users should be able to query the system to find out whether there are spaces available in a particular parking area. The system should be able to tell them whether there are spaces available in that area, and if there are, how many there are.
- The system should record whether the user is entitled to use a disabled or reserved parking spot.
- The system should undergo rigorous testing before launch.
- The system should be easy to use.

**Optional features**

- In addition to being able to query the system for information about a specified parking area, a user should be able to find the nearest location with available spots, and be provided with directions to it.

2. Review the new information below.

A colleague of yours has commenced design, and has produced (based on the results of requirements and analysis), a proposed UML class diagram (see last page of this specification).

A draw.io diagram will be made available if you want to re-use portions of this.

3. Decomposition. You realize your colleague's class diagram has made no effort to *decompose* the system into sub-systems: all the classes are at the "top level" of the system.

Based on the diagram, and from information obtained in the week 9 workshop, you should:

- propose a decomposition of the system into subsystems, and
- for each subsystem, describe briefly what services it provides.

You should aim to keep your discussion of decomposition to 2 pages or less, but this does not include space allocated to diagrams.

If providing diagrams of your decomposition (which is recommended) you may use either an appropriate UML diagram or an informal "block and line" diagram.

4. Decoupling. Attempt to identify at least one area of the system which suffers from *overly tight coupling*. Provide a class or object diagram of that portion of the system, and explain what you think the problem is. Suggest any improvements you would make.

Feel free to make suggestions which involve renaming, adding, or removing classes, or altering class attributes or relationships. If you use any *design patterns*, identify which ones you have used and why.

You should aim to keep your discussion of your design improvements to 3 pages or less, but this does not include space allocated to diagrams.

Note that based on your suggested improvements, you may wish to re-visit the decomposition from section (3) and alter it as needed.

5. Research and propose a strategy for handling persistent data in the system.

You should make suggestions for:

- what sort of technologies will be used for handling persistent data (for instance, files, relational databases, non-relational databases, or some combination of these)
- for each technology, provide:
  - a justification for why it is appropriate, and what alternatives you considered
  - an explanation of what entities from the design will be stored in it
  - a discussion of any risks or problems you think will need to be borne in mind when implementing persistent storage.

It is recommended that this section of your report be kept to 3 pages or less.

6. Write a short "Reflection" section ($\frac{1}{2}$ page to 1 page). You should reflect on on any aspects of the project your group found useful/surprising/difficult/etc.

7. Submit your report, containing sections for tasks 3–6.

You report should meet the following requirements:

- It should be in PDF format, and use A4 size pages. It should clearly show the name and student number of each member of the group.
- The font for body text should be between 9 and 12 points. The report should contain numbered headings, with useful heading titles. Any diagrams, charts or tables used must be legible and large enough to read when printed. All pages (except possibly the cover, if you have one) should be numbered.
- If you give scholarly references or citations of websites, you may use any standard citation style you wish, as long as it is consistent.
- Please ensure any references you give have an appropriate level of authority. For instance, if discussing design techniques or definitions, references to scholarly or industry literature are appropriate. If discussing particular storage technologies, references to the reference documentation are appropriate. In general, however, sources such as blog posts or wikipedia are not considered sufficiently authoritative.

You may wish to provide your decomposition diagrams as separate PNG or PDF files. (Since otherwise, they may be hard to read, if fitted into an A4, portrait-orientation page.) Submit these as (for instance) "decomposition-diagram-1.png" and "decomposition-diagram-1.png", and refer to them from your PDF report as needed.

Submit your files via `cssubmit`. Only one person per group should submit via cssubmit.

## Project discussion workshop

The week 9 workshop will be dedicated to discussing the project and providing additional information. Email Arran Stewart with 1-3 questions your group would like addressed, prior to the day of the workshop.

## Feedback

Each group has an opportunity for feedback on their project, by appointment with the lecturer any time *until* the week the project is due. That is, feedback appointments are available up to the end of week 10.

## Assessment

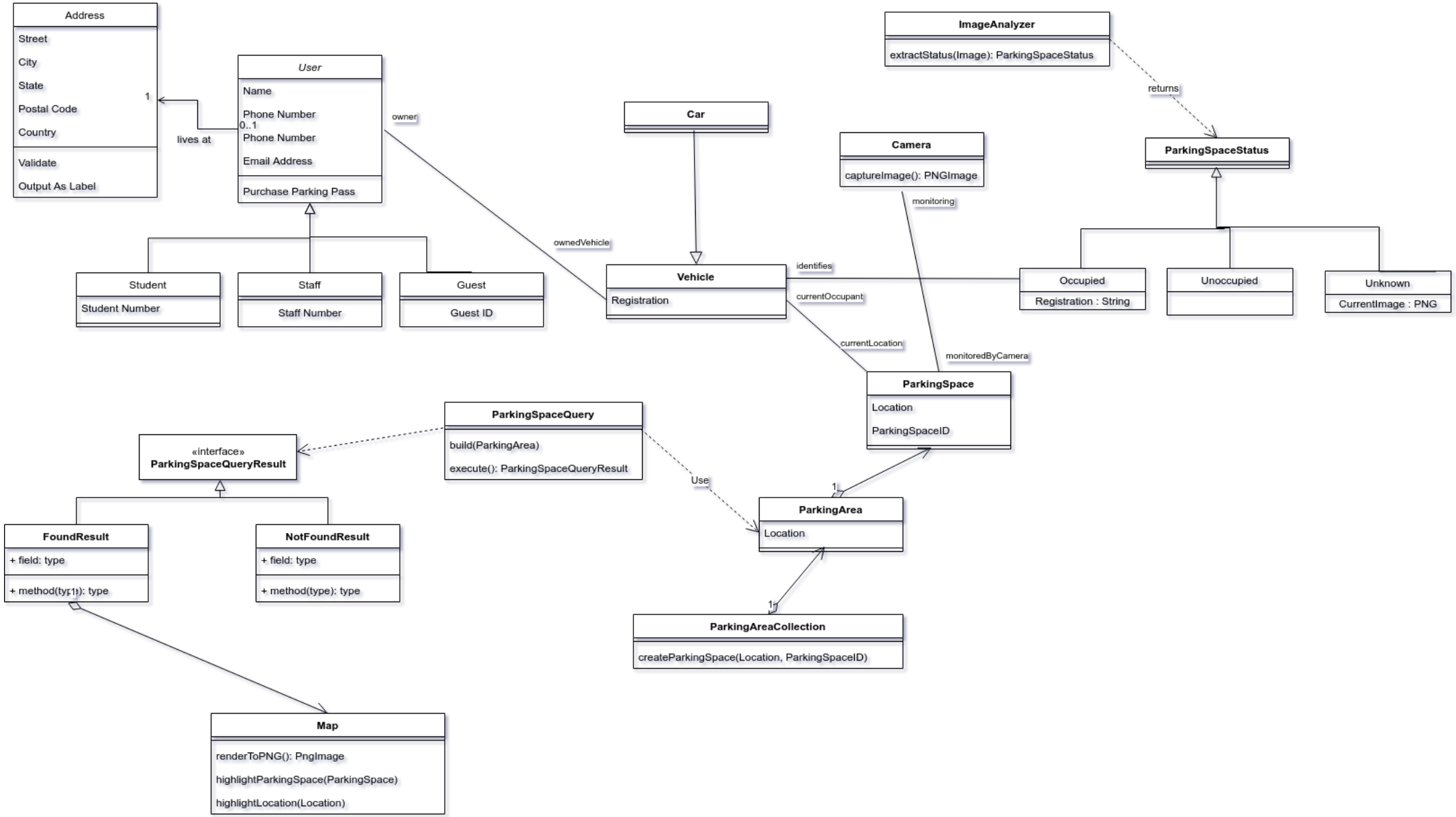
Each component of the report will be assessed on whether

- it is clear and logically laid out
- it meets the requirements given
- it describes and justifies any assumptions made.

The project will be marked out of 25 using the following marking criteria:

- Proposed decomposition: 10 marks.
  Criteria: Identifies main subsystems; justifies the decomposition; assumptions described and justified; describes service provided by subsystems; succinct: adheres to the page limit.
- Design improvements: 10 marks.
  Criteria: Identifies and explains problems with design; identifies and justifies suggested improvements; identifies any design patterns used.
- Persistent data proposal: 15 marks
  Criteria: Identifies and justifies use of relevant technologies; discusses alternatives; identifies entities stored; identifies and assesses risks or problems.
- Reflection: 2 marks
  Criteria: Succinct and insightful reflections on the project process
- Presentation: 2 marks
  Criteria: See project details above

Good luck!
Arran Stewart and Rachel Cardell-Oliver
CITS4401 lecturers


Address
Street
City
State
Postal Code
Country
Validate
Output As Label
1
lives at
0..1
User
Name
Phone Number
Phone Number
Email Address
Purchase Parking Pass
owner
ownedVehicle
Student
Student Number
Staff
Staff Number
Guest
Guest ID
Car
Vehicle
Registration
ImageAnalyzer
extractStatus(Image): ParkingSpaceStatus
returns
Camera
captureImage(): PNGImage
monitoring
ParkingSpaceStatus
identifies
currentOccupant
currentLocation
monitoredByCamera
Occupied
Registration : String
Unoccupied
Unknown
CurrentImage : PNG
ParkingSpace
Location
ParkingSpaceID
ParkingSpaceQuery
build(ParkingArea)
execute(): ParkingSpaceQueryResult
«interface»
ParkingSpaceQueryResult
Use
1
ParkingArea
Location
FoundResult
+ field: type
+ method(type): type
NotFoundResult
+ field: type
+ method(type): type
1
ParkingAreaCollection
createParkingSpace(Location, ParkingSpaceID)
Map
renderToPNG(): PngImage
highlightParkingSpace(ParkingSpace)
highlightLocation(Location)