ICT381 - Software Engineering

Assignment One – Software Project Analysis and Design
(Internal Students)

This assignment is to be done in groups of up to three students with a single entry handed in on both hard and soft copy. The complete assignment is worth 25% of the final mark for this unit.

A first draft is to be shown to the lecturer on or before 16th April, 2004 with a short formal presentation (including a one page information sheet), to the class during Week 10 Lecture/Workshop (13th May, 2004) on your approach. Descriptions of two suggested projects are given in the last section. (Students are free to suggest other project but it must be of comparable size and complexity. Students should consult lecturer for approval of alternative project.)

The work will be done in four stages:

(i) Discuss the work, consider ways of going about it and produce a list of members of the group with their roles and responsibilities. It is suggested that the team should include a team leader, a document coordinator, researchers, writers and reviewers.

(ii) Review existing material, text books, standards and relevant reference materials.

(iii) Produce a first draft of the report to be handed in (and in a form suitable for presentation to a tutorial) on 16th April, 2004 or earlier. Feedback will be provided to the team on Week 8 lecture, 29th May, 2004.

(iv) Attending to feedback from (iii) and produce a final copy of the report. The group is to hand in one complete printed copy of the report plus a copy on CD-ROM or floppy disk. The lecturer may retain this copy. Each member of the group is to have their own personal copy.

It is expected that the project should cover the following major activities:

- Determining clients needs (Feasibility, Systems analysis and Requirements Specification)
- Design of the solution (broad and detailed system design)
- Estimation on the time and resources required to implement the code and all other deliverables (including acquisition or reuse of externally produced components)
- Plan for delivery of the system (Integration of all parts, hardware, installation)
- Test plan for commissioning and ongoing maintenance.

For each activity, there should be assistance with estimation of time and resources, identification of milestones, assessment of possible risks.

Assessment Components of Assignment 1 are:

Draft Report (To be submitted on or before of Friday, 16th April, 2004) 10%
Final Report (To be submitted on or before Monday 10th May, 2004) 10%
Presentation (To be made on Thursday, 13th May, 2004) 5%
Suggested Software Projects:

1. **Automatic Surveillance and Monitoring system.**

**Description:** The project aims to develop a fleet of autonomous robots for surveillance purpose. It is intended to develop a system to control the robots over a wireless network. Each robot contains embedded software that interfaces with a set of sensors and actuators that allow the robot to navigate and to communicate with sensors around the premise. Surveillance video or images are to be taken at specific locations and upload to the central command station. Different levels of defensive measures such as electric shock or pepper spray are also included which can be activated to protect the robot. Decisions on the use of such measure are made by the “commander” except in cases of immediate threat. The system is supplied with a route map before the start of a round of patrol. Multiple robots are used and they are not necessarily doing the same round of patrol. In case of emergency, multiple robots can be summoned together to deal with the event.

The embedded software system is to be extended to interface with a traffic/event database over a wireless network. The database will maintain identification of the robots, traffic information, event information, road maps and location information. Due to memory restrictions the system must allow for the route to be downloaded in segments for bigger area of coverage.

**System requirements**

1. The onboard navigation system will access the information database in order to navigate the robot, and to response to any abnormal conditions.
2. The system will interact with the ‘commander’ through a keypad and a small LCD display.
3. At the start of a round of patrol, the ‘commander’ will be required to enter a valid PIN and an itinerary followed by a *start* command. The itinerary will comprise the check-points and any intermediate stopovers during the round of patrol.
4. The system will provide a continuous update of the robot’s location, surrounding conditions, and the expected time of arrival at the next check-point (throughout the journey).
5. The robot may periodically recommend route changes based on local conditions or request from other robots. The ‘commander’ may choose to accept or decline suggested route changes.
6. The ‘commander’ will be able to intervene to stop the robot by requesting a *normal* stop or an *emergency* stop. If a normal stop is requested, the robot will only stop when it is safe to do so.
7. Each robot will include a facility to request the use of different levels of defensive measures. However, in case of immediate threat, the robot will react in reflex action to use an appropriate level of defence.
8. The robot also include request for emergency assistance. All requests for emergency assistance will be transmitted to the Emergency Services Center which will respond by informing the commander an estimated time of arrival of the emergency crew.
9. If the system is unable to communicate with the database, then it must initiate a safe stop for the robot.
2. A Tourist Information System

Short description

The aim of the system is to provide users of information on public transport, local attractions, accommodation, “what’s on” events, restaurants and entertainment venues around Perth and the State of Western Australia. It should also advise the tourist the best way to reach their destinations, cost, and additional attractions along their selected route of journey. These systems will be located at strategic positions to maximise its utilisation and the information will be communicate to a central database using appropriate communication means.

Project Requirements

1. The system should advise the user the best solution.
2. You have to determine the stakeholders besides the users.
3. You have to determine where to access and gather the data.
4. You have to determine who will pay the system and to provide estimation on the cost of the project in terms of development and operation.
5. How will the system communicate to the user? to the database? to other stakeholders?
Deliverables
The following documents should be produced (you team will probably want to prepare them concurrently):
- a requirements definition based on the viewpoints of the stakeholders (3-6 pages),
- a rough architecture of the system, showing the over-all data-flow,
- a system requirements specification (3-6 pages).
- Accompanying the specification (as parts/appendices):
  - a glossary of abbreviations and terms (about 1-2 pages),
  - a sequence diagram showing that any event (like an accident) causes the system to propose an alternative itinerary to a user during travel.

Issues
- Proper organisation of the requirements definition and specification.
- Proper division between the requirements definition and specification.
- Traceability.
- Proof of prioritisation, conflict detection and resolution.
- Validity, proper focus on what the system does.
- Testability.
- Almost all stakeholders are represented.
- All kinds of requirements are represented.
- Proper use of pictures and associated text for the architecture/data-flow model and use case.

Non-issues
The following topics should be avoided, or dealt with very briefly. Not because they are uninteresting or unimportant, but because they are not part of the assignment (detailed design) or the course.
- Details of the user interface.
- Details of the data communication.
- Details of the hardware.
- Details on how to solving the problem of what is the optimal way from A to B given the current timetable.
- Solutions to problems of privacy and security.
- Completeness would require hundreds of pages. It is sufficient to provide 20 pages or so (depending on your font-size, density of the text and diagrams). So completeness in the usual sense is not an issue. However, as stated above, all kinds of requirements should be represented.

Remarks and hints
- Naturally, you will identify yourself with the user of such a system. But try to identify all other stakeholders and some of their requirements (to the extent your domain knowledge reasonably allows).
- Be concrete! Provide estimation on response times, names of standards, and the like. Take decisions! A bad decision is a sign of lack of domain knowledge. No decision is bad software engineering.
- Write verifiable requirements! When doubtful, describe how to verify the requirement.
- The glossary should define in an unambiguous way every term used in the document that might be unknown (like "POTS = Plain Old Telephone System") or ambiguous ("delay").