1. General Instructions

1.1 Project Goal

The goal of the semester project is to illustrate the work in a software development group, which applies the methods acquired during the course. The phases of the project include most of the typical phases in the development lifecycle of a software system. The scope of the project, however, matches the time frame of one academic semester.

1.2 The Phases of the Project

The project is executed in groups of four students each. There are five submissions in the project:

- Requirements (15%)
- System/Subsystem Requirements Specification (SSS) (20%)
- System Architecture + Software Architecture (SSDD+SAD) (15%)
- Software Requirements Specification + Software Design (SRS+SDD) (20%),
- Implementation and testing (30%).

In the first three phases the whole system is defined, designed and analyzed; in latter phases only a part of the system is designed in detail, implemented and tested.

The tasks in each phase should be distributed uniformly among the members of the group, but a fully cooperative teamwork is a key contributor to project success. Each submission should include a description of the responsibility of each member in executing the relevant phase. You may consider assigning one of the team mates as a project leader. The duties of a project leader will include coordinating meetings, distributing tasks, and deciding when a certain task is completed. The project leader should not limit his activities only to those written above. He should also assign a reasonable distribution of the engineering work onto himself.

The project employs object-oriented methodologies, using the UML language for the analysis and design phase and the Java language in the implementation phase. The RoboLib case study should serve as an example and format for the submissions. UML diagrams may be drawn using any graphical or CASE tools (e.g. enterprise architect, Visual Paradigm, Visio, office-graphics etc.). Attention should be paid, however, to use the UML 2.0 syntax correctly!
There is no mandatory requirement to use Microsoft software for producing the submission documents. Furthermore – students are encouraged to make use of Open Office tools.

1.3 Grades
Each phase has a different weight in the project's grade (which is 50% of the final grade – "TAKEF"). Each student in each group receives an individual grade by his/her responsibilities and contribution to accomplishing the objectives of the relevant phase.

Good luck!
2. Requirements Phase – 15%

The front page should include the following details:

- A unique group name (you may include your logo…)
- Names and ids of the group members
- Number of cell to return the submission

The submission should include a short summary of the distribution of the responsibilities among the group members, and requirements database, described below:

Requirements Data Base:
A table that includes the requirements derived from the project description and frequently asked questions list. The table should have the following columns: Id, Description, Source, Functional Category and Nonfunctional Category (as presented in the lectures and tutorials. You may use RoboLib's requirements table as a template). If some requirement is not consistent with the project description, indicate in the comments the number of answer in the frequently asked questions list, on which you base this requirement.

Good luck!
ePark – The 21st Century Family Fun

Client's Story

ePark is a fully computerized theme park, comprising a large number of exciting rides, some of which are defined as "extreme". All the rides are connected to a central control computer through a wireless network.

A child, arriving at the park with an adult guardian, registers and receives an electronic brace, to be worn on the wrist all the time. The brace serves both as an electronic ticket (eTicket) for the rides and as a means to indicate the location of the child in the park at any time. The child's age and height are recorded and loaded onto the eTicket, for ride entrance approval.

The guardian "recharges" the child's eTicket by purchasing entrances to the various rides. This is done using a password which the guardian receives upon registration. A child may enter a ride only when all the following conditions are met:

- Her electronic tickets contains a valid entrance to the ride
- She conforms to the minimum age and height constrains of that ride
- Her time limit has not expired
- The ride is in active and is standby for use.

In addition, an explicit guardian's prior approval is needed for each extreme ride.

The child can independently enjoy the park, while the guardian can monitor her activities, as described below, through the ePark's internet site, from anywhere. Alternatively, the guardian may stay at the ePark Café, using one of the guardian on-line workstations over the ePark's wireless network. The ePark map is displayed to the guardian and the child's icon, is shown on the map, updated every 30 seconds. One guardian may register and control more than one child. In this case all the children under his control are shown on the same map, by different icons. Double-clicking a child's icon will open the child's eTicket screen, in which the guardian can purchase or cancel entrances, limit the child's time, approve extreme rides etc. When the child's time limit has expired, a warning is displayed on the guardian's screen. The eTicket status is updated every time a child enters a ride.

A supervisor monitors and controls the entire park activities using a dedicated workstation, which is directly connected to the central computer. The supervisor is constantly updated about the status of all the rides in the park. The supervisor can define new rides and control the rides' parameters, including age and height constrains, capacity limit and activity (e.g. activate/deactivate rides according to forecasted use). Use 2-hour forecasting is displayed to the supervisor, based upon the total entrance purchasing. Every ride has a self-testing mechanism, which can change the ride's status automatically to out-of-order. Changing a ride back to in-order is done manually by the supervisor. Every change in the rides' status will be reported immediately to all on-line guardians.
A ride can be in-order or out-of-order, an in-order ride can be active or inactive, an active ride can be in standby or running. Each ride has an entrance gate and an exit gate. A ride starts up in standby mode with its entrance gate open and its exit gate closed. Children enter the ride through the entrance gate, which remains open for 5 minutes, or until the capacity limit of the ride is exceeded. Then the gate closes automatically and the ride is running. When the ride is over the exit gate opens automatically and the ride enters into standby mode. The system confirms that everybody exited the ride before closing the exit gate and opening the entrance gate. When the self-testing mechanism of the ride indicates a fault the ride stops automatically, the exit gate opens and the supervisor is reported accordingly.

A monthly report, which is produced for the ePark owners, contains statistics of use, income, ride load balance, returning visitors etc.