1. Goals
This phase focuses on detailed design to a part of the system. The build 1 of the system (that you will be required to implement) is specified at the end of this document as a highlighted text in the client story.

You are also required to submit the SSS document (with the requirement specification), SSDD and SAD documents. This part is not graded but is used to check that your design is consistent with the analysis. You may change and update previous documents to preserve this consistency.

This phase includes the software requirements specification and the software design document of the defined subsystem. The goal is to provide the software use case diagrams and specifications of the subsystem, class diagram of the subsystem and sequence diagrams of all software use cases you defined.

2. Responsibility Distribution
The tasks should be distributed uniformly among the members of the group. You are required to describe the responsibility of each member in executing the phase. Define the role and actual work of each member, in the level of functional responsibility: each member is responsible for some modules and/or functionality of the system.

3. Submission
This phase includes documents of software requirements specification (SRS) and software design document (SDD) for the build 1 of the system defined in this document. Cross reference table to the requirements must be provided in order to indicate consistency with the previous submission. The document (including previous phases) should be printed and submitted as one coherent document. The cover page should include the group name, and the names and ids of all group members.
Use Robolib SRS and SDD documents as references for the format of the submission (diagrams, tables etc.).

Pay attention, that if your implementation of the build 1 contains more than one CSCI (computer software configuration item), then you have to provide a separate SRS document for each item, as was described in the tutorial.

We recommend you to use UML tools for drawing deployment and component diagrams. There are number of the free-of-charge UML tools available on the web. In the faculty, you can use VisualParadigm tool, available in the SSDL library (2nd floor). All students in the course should have an ability to access this lab. The entrance to the lab is with your student card, and the account is your t2 account. If you have any problem to access the lab, please contact Dmitry (pidan@cs) or Roy (royl@cs).

Although the use of UML tool is just a recommendation, using a correct UML syntax (as was shown in lectures and tutorials) in deployment and component diagrams is a requirement.

Please inform Dmitry (pidan@cs) of any problem, and in particular Miluim, as soon as such issues arise.

### 3.1 Software Requirements Specification

The document should include the following artifacts for each software component:

- Use Case Diagram
- Use Case Specification table for each use case:
  - Use-case name and goal
  - Use-case stakeholders and their interests
  - Pre and post conditions
  - Trigger event
  - Main Success Scenario
  - Alternatives and exceptions

### 3.2 Software Design Document

The document should include the following artifacts:

- Class Diagram
- Class Description Table
  - Class name
  - Class members, their descriptions and reference to the requirements
- Sequence diagrams for all software use cases

The class model you provide should be somewhere in the middle between analysis level and design level class models. That is, you are not required to put in all classes you may use in your implementation (for example GUI details like Button, TextField etc.), however you are required to encounter in your class model every class that participates in your sequence diagrams.
3.3 Cross reference table

Extend the requirement table in the requirements document to include forward traceability columns, which indicate for each operational requirement and other relevant requirements the sequence diagram and the class that implements it (refer to Robolib requirements document ver.2). For the requirements that are not covered by this build 1, according to the highlighted text in the client story, just leave the additional columns empty.

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 guarian'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.