Components and Sequence Diagram

ARNON LAZERSON
Logical Architecture

- In the deployment diagram we identified the software artifacts installed in the various nodes.
- Each software artifacts is composed of functional components.
Functional Analysis

- Identify system functionality
- Group the functions
- Allocate functions to components
- Define logical interfaces
System functionality

The sources for system functionality are:

- Requirements Table
- Use Cases
- Activity diagram
### Component Diagram

#### Device C&C SW

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervisor I/F</td>
<td>Monitor and control park users</td>
</tr>
<tr>
<td>Device Control</td>
<td>Control and monitor equipment</td>
</tr>
</tbody>
</table>

#### DB Management SW

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Rep. Manager</td>
<td>Manage equipment database</td>
</tr>
<tr>
<td>User Rep. Manager</td>
<td>Manage user repository (children and elders)</td>
</tr>
</tbody>
</table>

#### User Management SW

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entrance Equip. Manager</td>
<td>Manage entrance equipment (turnstiles, measurement stations)</td>
</tr>
<tr>
<td>Usage Manager</td>
<td>Manage user usage in the park</td>
</tr>
<tr>
<td>Bracelet Control</td>
<td>Locate children using their bracelets</td>
</tr>
</tbody>
</table>

---

**COMPONENTS AND SEQUENCE DIAGRAM**
Sequence Diagram

- Describes a process as a sequence of actions
  - Interactions amongst functional entities
  - Interactions between functional entities and the environment

- How objects interact with one another and in what order
Life line and activation
Communication
Interaction frames

- Fragments in a sequence diagram describing nesting, control blocks etc.

- Frame types (operators):
  - **seq**: part of a sequence that can be performed independently
  - **ref**: reference to a diagram described elsewhere
  - **par**: parallel execution
  - **loop**: loop
  - **alt**: alternatives (switch case)
  - **opt**: optional execution (if ... then)
SUC-2

ניהול כרטיס אלקטרוני

COMPONENTS AND SEQUENCE DIAGRAM

Guardian (App)

Usage Manager

User Repository Manager

1.0 open eTicket(ID,PWD)

1.1 load eTicket()

1.2

1.3 display eTicket()

1.4 add/cancel entries()

ref
eTicket Update
SUC-3: יציאה מהפארק

COMPONENTS AND SEQUENCE DIAGRAM

Entrance Equipment Manager
Usage Manager
User Repository Manager
Guardian (App)
Entrance
CC Company

1.0 exit park()
1.1 take bracelet()
1.2 take bracelet()
1.3 Balance = finalize account(): int
1.4
1.5 Balance = finalize account(): int
1.7
1.6 debit CC(Balance)
1.8 delete eTicket()
1.9
SUC-4: מעקב אחר ילד

COMPONENTS AND SEQUENCE DIAGRAM

Guardian (App)

Usage Manager

Bracelet Control

Child (Bracelet)

1.0 track child(ID)

1.1 display park map()

1.2 Loc= locate child (ID): double

1.3 get location(): Loc

1.4

1.5

1.6 display child's icon()

1.7 [child in device]: display device icon()
SUC-5

COMPONENTS AND SEQUENCE DIAGRAM

Usage Manager
User Repository
Manager
Device Repository
Manager

alt Event
[guardian adds entries]
1.1 P= calculate price(): int
1.2 check guardian's balance(): boolean
1.3

opt Balance
[balance allows entries]
1.4 add entries to eTicket()
1.5 update guardian's balance()

[guardian removes entries]
1.6 remove entries from eTicket()

[child entered a device]
1.7 update - child in device()

[child exited a device]
1.8 check device status()
1.9

1.10 [device not broken]:
remove entry from eTicket()
1.11 update - child not in device()
COMPONENTS AND SEQUENCE DIAGRAM

1.0 enter device(ID)

1.1 get device data(): DeviceData

1.2

1.3 get eTicket Data(): eTicketData

1.4

1.5 check child's eligibility(): boolean

1.6 open()

1.7 eTicket update (child entered device)

opt [child eligible to enter]

ref eTicket Update

Device Control

Device Repository
Manager

User Repository
Manager

Usage Manager

Device (Gate)

Child (Bracelet)
COMPONENTS AND SEQUENCE DIAGRAM

1.0 exit device(ID)

1.1 open gate()

1.2

1.3 eTicket update
(child exits device)

eTicket Update

ref

Supervisor «GUI» Supervisor I/F

Device Repository Manager

loop Device Definitions Approval
[until definitions approved]

1.0 show device(DeviceID)

1.1 get device data(): DeviceData

1.2

1.3 display device setup dialog()

2.0 setup device(DeviceID)

2.1 set device data(Definitions): Validation

2.2 approve definitions(): boolean

2.3

2.4 notify aproval/rejection()