

# System Requirements: DigiMergo

---

*PUM 2014*

DOCUMENT VERSION: 1.0

**Shall:** Must be included in the software released at end of project. Priority 1.  
**Should:** Will be done after all the “shall” requirements if there is time. Priority 2.  
**If time permits (ITP):** These requirements are features that are planned for later release or are experimental – build only if there is plenty of time left. Priority 3.

# Changes

---

## Version 1.0

Added Terminology section

### Moved up or new

- New specifying requirement and note: S.R. 4.5
- S.R. 6.4: Use a time-line metaphor
- S.R. 25: moved from *should* to *shall*.
- S.R. 34: moved from 26
- Logging: 24.3.1, 24.1.1, 24.5.4, 12.2

### Moved down or removed

- 3D not required: (old S.R. 3 – 3.3 & 12.2) moved from *Shall* to *If time permits*
- Removed “hideable” server config. (old S.R. 20.3)
- Removed S.R.2.1.1

### Terms and wording

- “Exercise” used consistently instead of “game”
- Formulation: S.R.2.1
- Formulation: S.R.4
- Formulation: S.R.6
- Formulation: S.R.18
- Formulation: S.R.19
- Formulation: S.R.20
- Formulation: S.R.24.4

# Scenario editor

---

*Standalone application for building scenarios and resource packages.*

## Shall

- S.R.1:** (v.1) Have ability to create, edit and save scenarios as well as resource packages.
  - 1.1: Finished scenario will be saved to file (xml?)
  - 1.2: Finished resource package will be saved to file (xml?)
- S.R.2:** (v.1) Separate scenario creation & editing from scenario execution (running the exercise), since exercise clients may change in the future.
  - 2.1: Scenario file shall include information on what type of Exercise client it's designed for/playable on.
- S.R.3:** (v.1) Build 2D views with the ETS look and feel.
  - 3.1: All views, including meta views (transport times, overviews)
    - 3.1.1 New view "Genom-vindrutan": 2D camera image, film or Google Maps street view (or similar)
  - 3.2: Include possibility for adding custom graphics (e.g. images for accident scene)
- S.R.4:** (v.1.1) Be possible to place scenario on a map
  - 4.1: Calculate driving times using the map
  - 4.2: Manually place resources on map (e.g. location of ambulances at exercise start, location of hospitals)
  - 4.3: Possibility to specify pre-defined location for "brytpunkt"
  - 4.4: Possibility to specify pre-defined location for "genom-vindrutan-rapport"
  - 4.5: Genom-vindrutan view can include an optional map over accident site with "brytpunkt" and "genom-vindrutan" locations marked.

*Note: binding a scenario to a map (location) does not necessarily affect any other of the client views, rather it sets driving times and available resources (hospitals etc.) and their location.*

- S.R.5:** (v.1) Be possible to create scenario without using map
  - 5.1: Driving times and distances must be entered manually
- S.R.6:** (v.1.0) Have a script (manus) module, where instructor defines Events and general instructions for what will happen during the exercise.
  - 6.1: General instructions (körschema): textual description of what will happen during the exercise (similar to a movie script)
  - 6.2: Events that will occur at given time in a given order
    - 6.2.1 Suggested event behaviors: automatic, prompted
    - 6.2.2 Event type suggestions:
      - Environment: fire spreads, fire contained, gas leaks, building collapse
      - Resource: national resources arrive, other organization arrive, transport breaks down/delayed
      - Patient: death, health change, leaves area (if can walk)
  - 6.3: Pre-define Jump-points in time for long scenarios. Defines state and position of all patients, environment, and resources for this given time.
  - 6.4: Use a time-line metaphor
- S.R.7:** (v.1) Be able to set default resource package to be used for a specific scenario
- S.R.8:** (v.1) Use a keyboard and mouse interface.

## Should

- S.R.9:** (v.1) Script (manus) should be exportable to pdf for printing and used as a physical tool during exercise.

## If time permits

- S.R.10:** (v.0.5) Scenario forks: have more than one pre-defined sequences of events that the instructor can choose from during play.
- S.R.11:** (v.0.5) Build 3D views for scenarios
  - 11.1: Accident site views (multiple or single sectors), “uppsamlingsplats” site
    - 11.1.1 New view “genom-vindrutan-rapport”: view of accident site from defined location in 3D environment.
  - 11.2: Include 3D models of buildings, objects of interest (cars, trees, etc), and patients
    - 11.2.1 3D models of environment can be modified to indicate danger or damage (accident), e.g. burning building, crashed car, poison gas.
  - 11.3: Patients can be placed inside a bus, or by a burning building, etc. in order to create a realistic scene.

# Exercise client

---

*The actual simulators. Existing clients utilize touchscreens to simulate ETS whiteboards.*

## Shall

- S.R.12:** (v.1) Visualize "genom-vindrutan-view": surveying the accident site from the ambulance.
  - 12.1: 2D version: custom image, video or modified "Street view" (Google) connected to a map.
  - 12.2: Log when displayed, log when closed.

## Should

- S.R.13:** (v.1) Change (or improve) flip interaction
- S.R.14:** (v.1) Change (or improve - clarify) the interaction to transport patients of (and resources) from one view to another.
- S.R.15:** (v.1) Change (or improve) the representation of "incoming" objects on a view

## If time permits

- S.R.16:** (v.0.5) Build accident scene client in the 3D environment (how to interact? – touchscreen not great in three dimensions).
- S.R.17:** "Genom-vindrutan" view 3D version: camera view of the 3D environment – accident site

# Exercise manager

---

*Instructors' main tool for controlling a training exercise. Manages server, connected clients, scenario & exercise control, logging, and result visualization (after action review).*

## Shall

**S.R.18:** (v.1) Be a standalone application running on a separate PC that will be able to select Scenario, start and control Exercise.

*Note: will be used from setup to finish of an exercise, thus the interface must be segmented to support the current instructor task. Previous segments should never be lost/removed.*

**S.R.19:** (v.1) Separated Exercise Manager gui and server functionality. In the future we would like to control (parts of) the Exercise via remote devices (phones, tablets, etc).

**S.R.20:** (v.1) Possible to use the same Exercise Manager and server with new clients that will be explored in the future. Access Server via public API usable in future clients.

**S.R.21:** (v.1.1) Control server settings and status

21.1: View connected clients

21.2: See server information (addresses, ports, etc.)

**S.R.22:** (v.1) Setup up and specify training exercise

22.1: Choose scenario-file to be used

22.1.1 View information on required (recommended) number of displays for scenario.

22.2: Choose resource package (scenario may suggest default)

22.3: Choose exercise "strictness". What participants are allowed to do – e.g. can a patient be moved while undergoing treatment. In some exercises low strictness is appropriate.

**S.R.23:** (v.0.5) Monitor ongoing exercise

23.1: Select and display any Exercise client's view in miniature.

23.2: Display script (manus)

23.3: Display overview of current scenario status (number of patients on accident site, arrived at hospital, resource utilization, etc)

23.4: View live log data – filters required.

**S.R.24:** (v.1.0) Control ongoing exercise

24.1: Start, pause, resume and stop exercise

24.1.1 Log these events

24.2: Display script (manus)

24.3: Manage which view is displayed on which client.

24.3.1 Log instructors changes to views

24.4: Change exercise-time

24.4.1 Move to pre-defined Jump-Point – current state will be overridden by state defined in Jump-Point

24.4.2 If no pre-defined Jump-Point exist in scenario, instructors may manually move patients and resources in response to new time

24.4.3 Log time changes

24.5: Control events

24.5.1 Skip pre-defined events in scenario

24.5.2 Respond to prompted events

24.5.3 Create new manual events (inspel)

24.5.4 Log all events (even skipped)

**S.R.25:** (v.1.0) Include a control for quality indicators during exercise. Instructors will manually check and comment on whether each indicator was correctly achieved within given time-frame.

25.1: Log when checked and commented

**S.R.26:** (v.1) Display after action review (see result visualizer)

**S.R.27:** (v.1) Use mouse and keyboard interaction

## Should

**S.R.28:** (v.0.5) Be able to preview scenarios: information on accident site, type of client intended, views, patients, resources, and textual description of scenario. If time permits

**S.R.29:** (v.0.5) Build tablet controller (android or windows)

29.1: Use tablet device to check quality indicators and log these checks

29.2: Take live notes -> logged

# Result visualizer

---

*View within Exercise manager (or standalone application) which presents results as graphs and numbers for immediate after-action review.*

## Shall

- S.R.30:** (v.1) Have graph displaying number of patients on accident scene over time
- S.R.31:** (v.1) Have graph displaying number of triaged patients on scene (how many red, yellow, green) over time
  - 31.1: Visualize where they were thought the exercise
- S.R.32:** (v.0.5) Data which displays number of possibly avoidable complications & casualties
- S.R.33:** (v.1) Be easily projected or displayed on large screen for group discussion.
- S.R.34:** Export log data to excel file (existing functionality, update to match Result Visualizer)

## Should

- S.R.35:** (v.1) Display quality indicators and whether they were fulfilled

## Is time permits

- S.R.36:** (v.1) Graph visualization selection (line, bar, pie, etc.)



# Overarching Guidelines

---

*Guidelines are not explicit requirements, but should be used as guiding lights when developing.*

## Language and localization

Use English in gui, code and documentation.

If time permits: use localization files for easy translation.

## Code

Implement in WPF C#, using Visual Studio

GIT-versioning

## Update

Update logger: newly implemented features will be logged.

## Design

Simplicity is key - simple rather than “cool”

Flexibility – for instance, scenario editor must not depend on the any particular Exercise client. Exercises will in the future be, partially, controlled with portable devices – phones, tablets.

# Terminology

---

<i>Exercise</i>	Functional exercise in disaster medicine, using for the purpose of this document DigiMergo
<i>Instructor</i>	The trainers arranging and controlling the exercise
<i>Player</i>	Participants (learners) in an Exercise.
<i>Exercise Client</i>	Software used by Players' to interface with the system
<i>Exercise Server</i>	Software running a Scenario and connections to clients
<i>Exercise Manager</i>	Software used by Instructors' to setup and manage Exercise
<i>Scenario Editor</i>	Software used offline to create and edit Scenarios
<i>View</i>	What will be displayed by a particular Exercise Client (e.g. accident site, transport table)
<i>Log</i>	A time stamped history of everything occurred during Exercise; Player interaction, Instructor interaction, Events, Patient outcome
<i>Scenario</i>	Contains information on accident, Patients, Events and which Views that will be included in a Exercise
<i>Event</i>	Pre-defined occurrence in a Scenario which can be executed automatically or manually. For instance environment change or resource change.
<i>Jump-Point</i>	Pre-defined state in Exercise time