COPENHAGEN SCHOOL OF
DESIGN AND TECHNOLOGY

Date: xx-xx-2021

Project Repport
/ Project Title */*

DATxxx

Group:

/ Group Name or Nr. */*

Project Participants:

/ Name A + Birthday */*

/ Name B + Birthday */*

/ Name C + Birthday */*

/ Name D + Birthday */*

Document Version: 00.01.00

Document Template version: 00.04.03

Contents

/* TO BE REMOVED…

Remember to edit or delete all comment sections marked /* */

“Optional” sections – Do you need them?

Remember to update Contents.

*/

Document Template History

Date	Version	Author	Description
201x-xx-xx	00.01.00	HAAM	Initialized
2019-09-25	00.02.00	HAAM	Section added SS & FURPS cells
2019-11-19	00.02.01	HAAM	Added Operation Contracts
2020-03-09	00.03.01	HAAM	Re-ordered and "{}" replaced with "/* */"
2021-10-11	00.04.00	HAAM	Sections SSD, DCD, SD added.
2021-11-03	00.04.01	HAAM	Section DM added.
2021-11-15	00.04.02	HAAM	Section added: GRASP & Review
2021-11-16	00.04.03	HAAM	Section added: Package Diagram, State Machine Diagram & UP Iteration Plan

Document Version

/* TO BE REMOVED…

- Who has last updated and for what reason?
- What has changed?

*/

Date	Version	Author	Description
202x-xx-xx	00.01.00	xxxx	Initialized

Vision

/* TO BE REMOVED…

The GOAL!
It defines the stakeholders view of the product/service to be developed, specified in terms of the stakeholders key needs and features. Containing an outline of the envisioned core requirements.

Summary of System Features

Definition
Features are behavioral functions a system can do. They should pass this linguistic test:
The system does <feature X>.

For example:
The system does payment authorization.

The major features include:

- POS services
- Inventory management
- Web-based shopping
- ...

Other Requirements in the Vision?
In the Vision, system features briefly summarize functional requirements often detailed in the use cases. Likewise, the Vision can summarize other requirements (for example, reliability and usability) that are detailed in the Supplementary Specification. But be careful to avoid going down the path of repeating yourself.

See Larman for more information:
Chapter 7.6. NextGen Example: (Partial) Vision
Chapter 7.7. Commentary: Vision

*/

Business Rules / Domain Rules

/* TO BE REMOVED…

Can be part of the Supplementary Specification (SS)

Typically describe requirements or policies that transcend one software project—they are required in the domain or business, and many applications may need to conform to them. An excellent example is government tax laws. Domain rule details may be recorded in the Supplementary Specification, but because they are usually more enduring and applicable than for one software project, placing them in a central Business Rules artifact (shared by all analysts of the company) makes for better reuse of the analysis effort.

See Larman for more information:
Chapter 5.5. How are Requirements Organized in UP Artifacts?
Chapter 7.10. NextGen Example: Business Rules (Domain Rules)
Chapter 7.11. Commentary: Domain Rules

*/

Requirements

/* TO BE REMOVED…

Extracted from meeting, interview and/or other gathered information for the new system

- Req001…
- Req002…
- Req003…
- …

*/

Supplementary Specification

/* TO BE REMOVED…

Elements of the Supplementary Specification include:
- FURPS+ requirements—functionality, usability, reliability, performance, and supportability
- reports
- hardware and software constraints (operating and networking systems, ...)
- development constraints (for example, process or development tools)
- other design and implementation constraints
- internationalization concerns (units, languages)
- documentation (user, installation, administration) and help
- licensing and other legal concerns
- packaging
- standards (technical, safety, quality)
- physical environment concerns (for example, heat or vibration)
- operational concerns (for example, how do errors get handled, or how often should backups be done?)
- application-specific domain rules
- information in domains of interest (for example, what is the entire cycle of credit payment handling?)

Use Either FURPS+ or Functional/Non-Functional
basically, everything not in the use cases. This artifact is primarily for all non-functional requirements, such as performance or licensing. It is also the place to record functional features not expressed (or expressible) as use cases; for example, a report generation.

See Larman for more information:
Chapter 5.4. What are the Types and Categories of Requirements?
Chapter 5.5. How are Requirements Organized in UP Artifacts?

*/

/* TO BE REMOVED…

!!! ONLY DO FURPS or Functional/Non-Functional, NOT BOTH !!!

*/

FURPS

Functional

/* To be removed — features, capabilities, security. */

Id	Requirement	Comment
FR001	…	
FR002	…	

Usability

/* To be removed — human factors, help, documentation. */

Id	Requirement	Comment
UR001	…	
UR002	…	

Reliability

/* To be removed — frequency of failure, recoverability, predictability. */

Id	Requirement	Comment
RR001	…	
RR002	…	

Performance

/* To be removed — response times, throughput, accuracy, availability, resource usage. */

Id	Requirement	Comment
PR001	…	
PR002	…	

Supportability

/* To be removed — adaptability, maintainability, internationalization, configurability. */

Id	Requirement	Comment
SR001	…	
SR002	…	

(FURPS)+

Implementation

/* To be removed — resource limitations, languages and tools, hardware, ... */

Interface

/* To be removed — constraints imposed by interfacing with external systems. */

Operations

/* To be removed — system management in its operational setting. */

Packaging

/* To be removed — for example, a physical box. */

Legal

/* To be removed — licensing and so forth. */

Functional Requirements

Id	**Requirement**	**Comment**
FR001	…	
FR002	…	

Non-Functional requirements

Id	**Requirement**	**Comment**
NFR001	…	
NFR002	…	

Domain Model

/* TO BE REMOVED…

A domain model is the most important—and classic—model in OO analysis.
Motivation: Lower Representational Gap with OO Modeling

Objectives

- Identify conceptual classes related to the current iteration.
- Create an initial domain model.
- Model appropriate attributes and associations.

See Larman for more information:
Chapter 9. Domain Models

*/

Use Cases

/* TO BE REMOVED…

- Description of each Actor and Goals
  See Larman for more information: Chapter 6.14. Guideline: Take an Actor and Actor-Goal Perspective
- Short Use Case Model description
- Use Case Diagram(s)
- Use Case 1 text – Brief/Casual/Fully Dressed
- Use Case 2 text – Brief/Casual/Fully Dressed
- Use Case 3 text – Brief/Casual/Fully Dressed
- …

Remember to add
“: INCLUDE *UCxx IncludedUseCaseName*.” in main use case step where INCLUDE is used.

Supporting **Use Case sections** template created by Alistair Cockburn

Use Case Section	**Comment**
Use Case Name	Start with a verb.
Scope	The system under design.
Level	“user-goal” or “subfunction”
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, and worth telling the reader?
Success Guarantee	What must be true on successful completion, and worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.

See Larman for more information: Chapter 6.9. What do the Sections Mean?

See Larman for more information:
Chapter 6. Use Cases

*/

System Sequence Diagram

/* TO BE REMOVED…

Objectives
• Identify system events.
• Create system sequence diagrams for use case scenarios.

See Larman for more information:
Chapter 10. System Sequence Diagrams

***/**

Operation Contracts

/* TO BE REMOVED…

Definition: What are the Sections of a Contract?

A description of each section in a contract is shown in the following schema.

Operation	Name of operation, and parameters
Cross References	Use cases this operation can occur within
Preconditions	Noteworthy assumptions about the state of the system or objects in the Domain Model before execution of the operation. These are non-trivial assumptions the reader should be told.
Postconditions	This is the most important section. The state of objects in the Domain Model after completion of the operation.

See Larman for examples: 11.9. Example: NextGen POS Contracts

See Larman for more information:
Chapter 11. Operation Contracts

*/

Package Diagram

/* TO BE REMOVED…

***/**

Design Class Diagram

/* TO BE REMOVED…

The UML includes class diagrams to illustrate classes, interfaces, and their associations. They are used for static object modeling.

See Larman for more information:
Chapter 16. UML Class Diagrams

***/**

Sequence Diagram

/* TO BE REMOVED…

Guideline
Spend time doing dynamic object modeling with interaction diagrams, not just static object modelling with class diagrams.

See Larman for more information:
Chapter 15. UML Interaction Diagrams

***/**

GRASP

/* TO BE REMOVED…

How GRASP was used?
Explain how following patterns was considered and used in the application

- Creator
- Information Expert
- Low Coupling
- Controller
- High Cohesion
- Pure fabrication
- Polymorphism
- Indirectiono
- (Protected variations)

See Larman for more information:
Chapter 17. GRASP: Designing Objects with Responsibilities

***/**

State Machine diagram

/* TO BE REMOVED…

***/**

"Optional" – Other Information?

/* TO BE REMOVED…

Other informative text/diagram to include?

*/

Glossary

/* TO BE REMOVED…

Use either 1 Glossary with 2 section or 2 Glossaries:
One for Domain/Business
One for Technical/Data dictionary

See Larman for more information:
Chapter 7.8. NextGen Example: A (Partial) Glossary for more information
Chapter 7.9. Commentary: Glossary (Data Dictionary)

***/**

Unified Process - Iteration Plan

Iteration	**Disciplines**	**Artifact**	**Reflection (After each iteration)**
I1 - Inception Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		
E1 - Elaboration Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		
E2 - Elaboration Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		
C1 - Construction Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		
C2 - Construction Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		
xx - x Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		

Test Cases

/* TO BE REMOVED…

Insert your Test Cases Table

***/**

REVIEW

Review ID	Artifact name (step 1):	Focus on (step 1):	Review note (step 2):	Correction/conclusion after review (step 3):