SRS and Project plan

ECE493-Lab

Scope

CMS is a web application that will support managing a conference organization process. This system includes registration, paper submission, peer reviewer assignments, publishing announcements, scheduling the accepted papers. CMS is going to be a conference management system for a technical specific conference hold by the XYZ department at University of Alberta, but it should be usable and extendable for other conferences. This version of CMS excludes of accommodation facilities for the conference audience. However, it provides useful information through announcement in the webpage. Connecting to the publisher and delivering the right of authors to the publisher is one of the tasks of conference management system, which is performed manually at this time and it is out of the scope of CMS.

Users

Check it again after writing system features’ scenarios

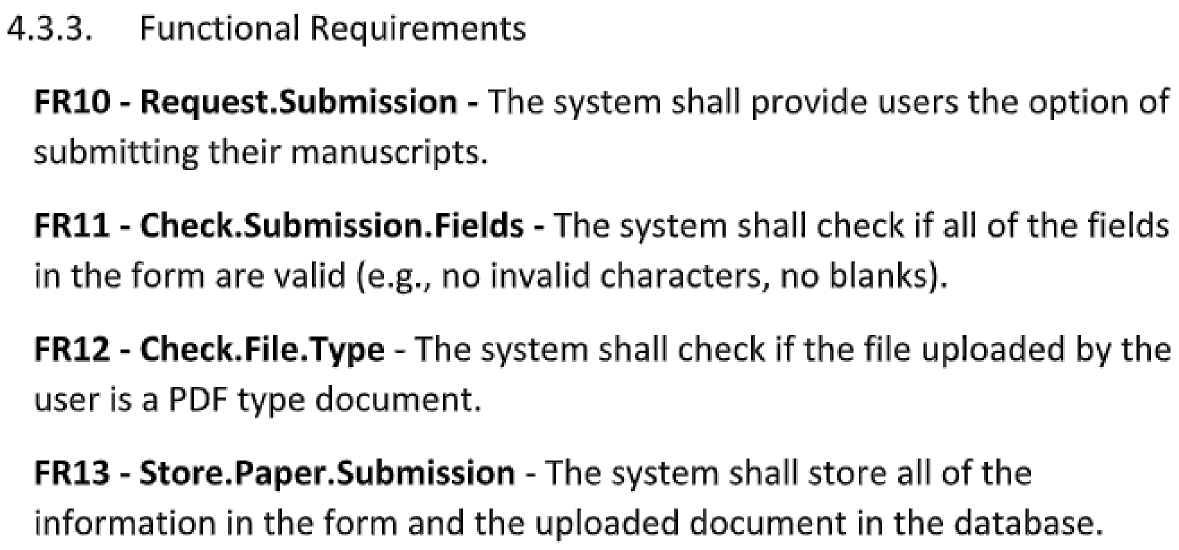
User Characteristics CMS is mainly meant to be used by the University of Alberta staff and its conference audience. However, it should be implemented in a way to allow future extensions in scaling the number and type of users. Users of the current version are:

- Public users: guest who visit the web page of CMS.
- Authorized users: registered users who are a referee, an author, or an audience who register for attending to the conference. This can be including:
    - Administrator: A university of Alberta member who mange the system.
    - Editor: This role is the same with the administrator in the current version of CMS but it can be changed in the future. Editor is in charge for managing the referee tasks and making decision about the papers.
    - Referee or reviewer: This role is an authorized member who can access to the list of assigned review tasks under her/his name.
    - Authors: Authors are corresponding authors who submit a paper to the system.

Software Requirements

- Write a scenario in software language for each business need.
- Split it to the related components
- It’s a practice

Paper Submission Feature:



4.3.3. Functional Requirements

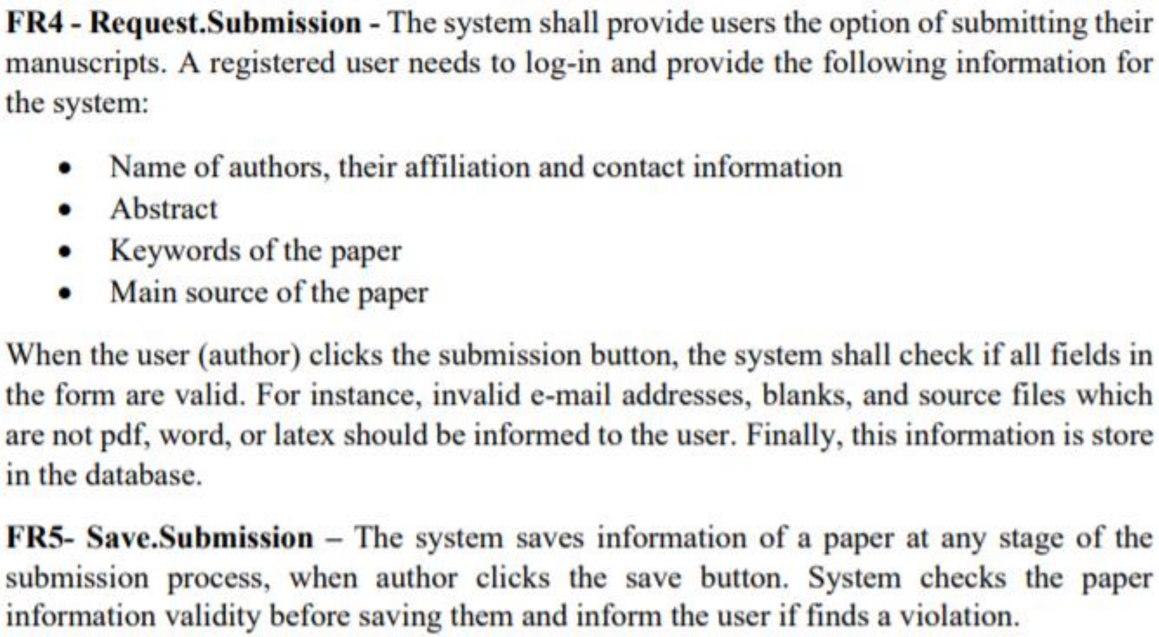
FR10 - Request.Submission - The system shall provide users the option of submitting their manuscripts.

FR11 - Check.Submission.Fields - The system shall check if all of the fields in the form are valid (e.g., no invalid characters, no blanks).

FR12 - Check.File.Type - The system shall check if the file uploaded by the user is a PDF type document.

FR13 - Store.Paper.Submission - The system shall store all of the information in the form and the uploaded document in the database.

A



FR4 - Request.Submission - The system shall provide users the option of submitting their manuscripts. A registered user needs to log-in and provide the following information for the system:

- Name of authors, their affiliation and contact information
- Abstract
- Keywords of the paper
- Main source of the paper

When the user (author) clicks the submission button, the system shall check if all fields in the form are valid. For instance, invalid e-mail addresses, blanks, and source files which are not pdf, word, or latex should be informed to the user. Finally, this information is store in the database.

FR5- Save.Submission – The system saves information of a paper at any stage of the submission process, when author clicks the save button. System checks the paper information validity before saving them and inform the user if finds a violation.

B

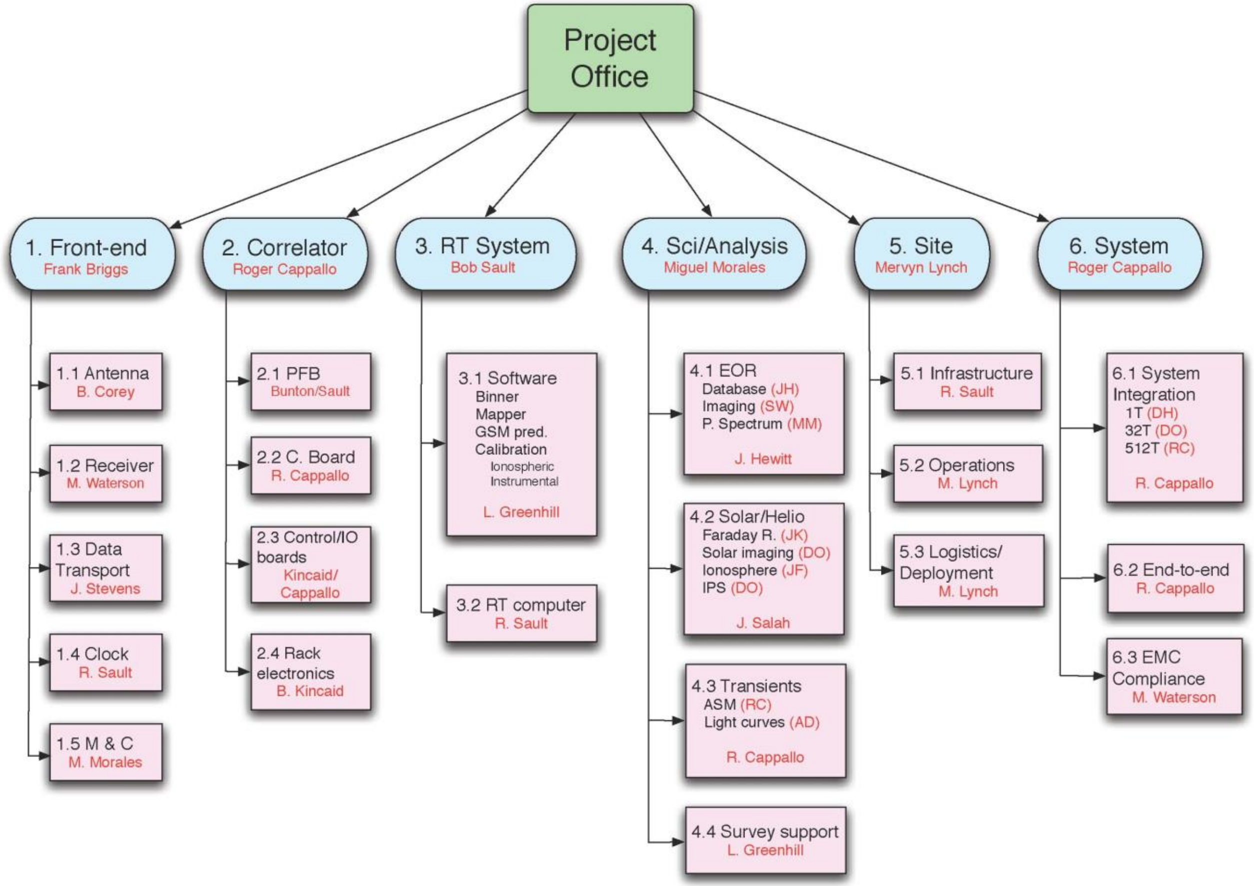
- Use a Software Language

Stakeholders

- Users
- Project manager
- Project owner
- Development team
- Suppliers
- Involved organizations
- etc.

WBS

What’s it look like?



Project Office

1. Front-end
Frank Briggs

1.1 Antenna
B. Corey

1.2 Receiver
M. Waterson

1.3 Data Transport
J. Stevens

1.4 Clock
R. Sault

1.5 M & C
M. Morales

2. Correlator
Roger Cappallo

2.1 PFB
Bunton/Sault

2.2 C. Board
R. Cappallo

2.3 Control/IO boards
Kincaid/ Cappallo

2.4 Rack electronics
B. Kincaid

3. RT System
Bob Sault

3.1 Software
Binner
Mapper
GSM pred.
Calibration
Ionospheric
Instrumental
L. Greenhill

3.2 RT computer
R. Sault

4. Sci/Analysis
Miguel Morales

4.1 EOR
Database (JH)
Imaging (SW)
P. Spectrum (MM)
J. Hewitt

4.2 Solar/Helio
Faraday R. (JK)
Solar imaging (DO)
Ionosphere (JF)
IPS (DO)
J. Salah

4.3 Transients
ASM (RC)
Light curves (AD)
R. Cappallo

4.4 Survey support
L. Greenhill

5. Site
Mervyn Lynch

5.1 Infrastructure
R. Sault

5.2 Operations
M. Lynch

5.3 Logistics/ Deployment
M. Lynch

6. System
Roger Cappallo

6.1 System Integration
1T (DH)
32T (DO)
512T (RC)
R. Cappallo

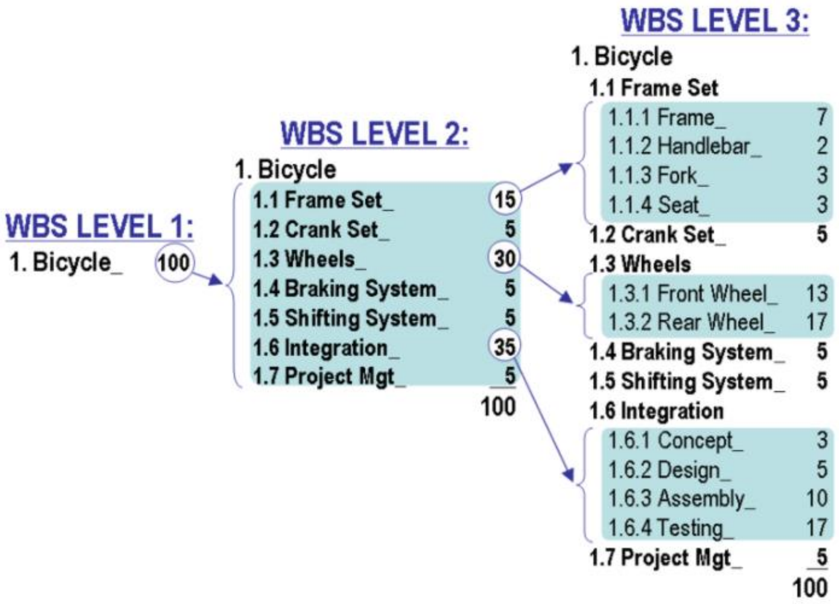
6.2 End-to-end
R. Cappallo

6.3 EMC Compliance
M. Waterson

WBS [PMBOK]

- Work Breakdown Structure (WBS) is usually a project management practice in which the project is break to the smaller tasks.
- A deliverable-oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables.
- A Work Breakdown Structure (WBS) is a hierarchical structure of things that the project will make or outcomes that it will deliver”.

Concept of tasks when make a bicycle:



WBS LEVEL 1:

1. Bicycle_ 100

WBS LEVEL 2:

1. Bicycle
1.1 Frame Set_ 15
1.2 Crank Set_ 5
1.3 Wheels_ 30
1.4 Braking System_ 5
1.5 Shifting System_ 5
1.6 Integration_ 35
1.7 Project Mgt_ 5
100

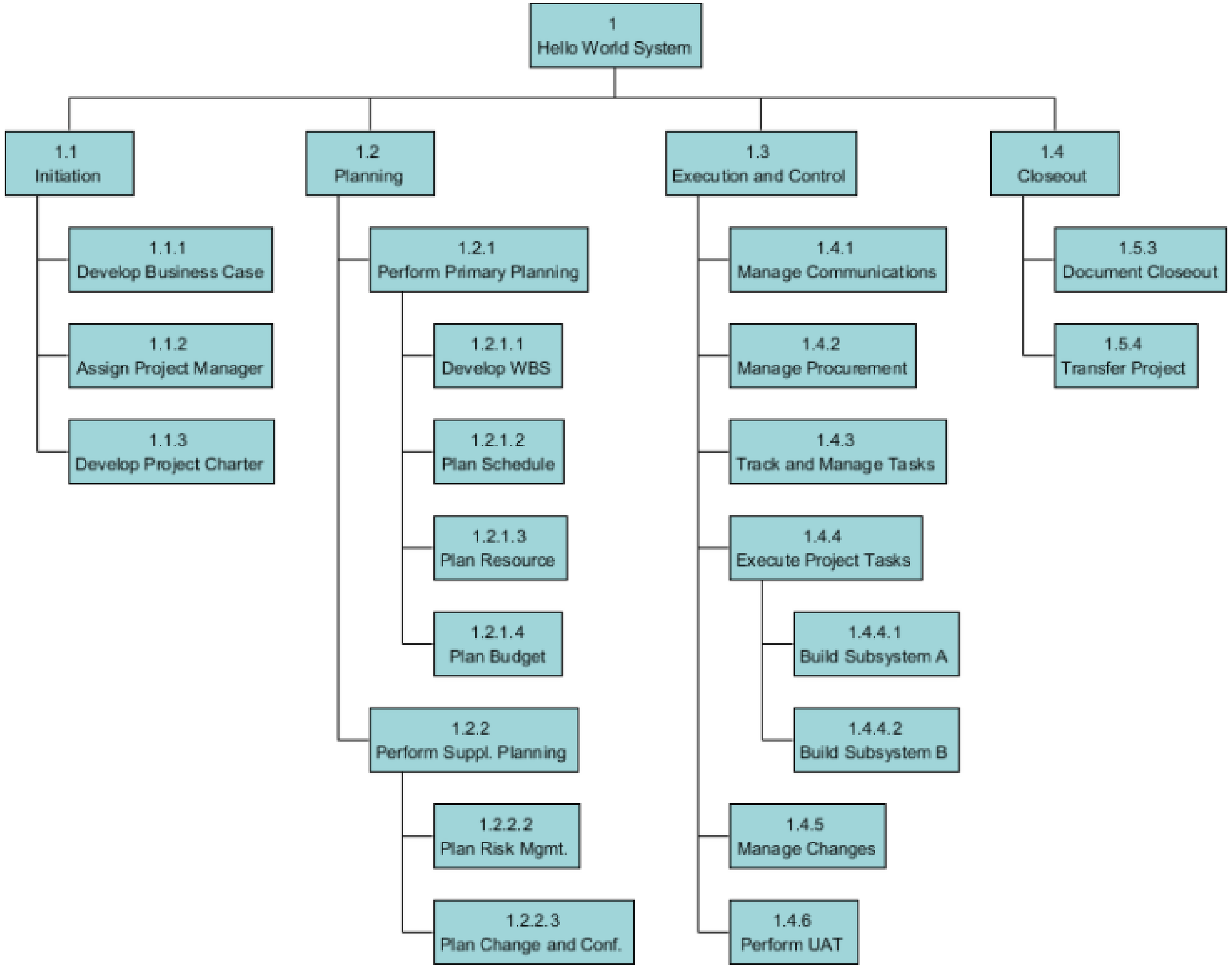
WBS LEVEL 3:

1. Bicycle
1.1 Frame Set
1.1.1 Frame_ 7
1.1.2 Handlebar_ 2
1.1.3 Fork_ 3
1.1.4 Seat_ 3
1.2 Crank Set_ 5
1.3 Wheels
1.3.1 Front Wheel_ 13
1.3.2 Rear Wheel_ 17
1.4 Braking System_ 5
1.5 Shifting System_ 5
1.6 Integration
1.6.1 Concept_ 3
1.6.2 Design_ 5
1.6.3 Assembly_ 10
1.6.4 Testing_ 17
1.7 Project Mgt_ 5
100

https://www.workamajig.com/blog/guide-to-work-breakdown-structures-wbs

WBS

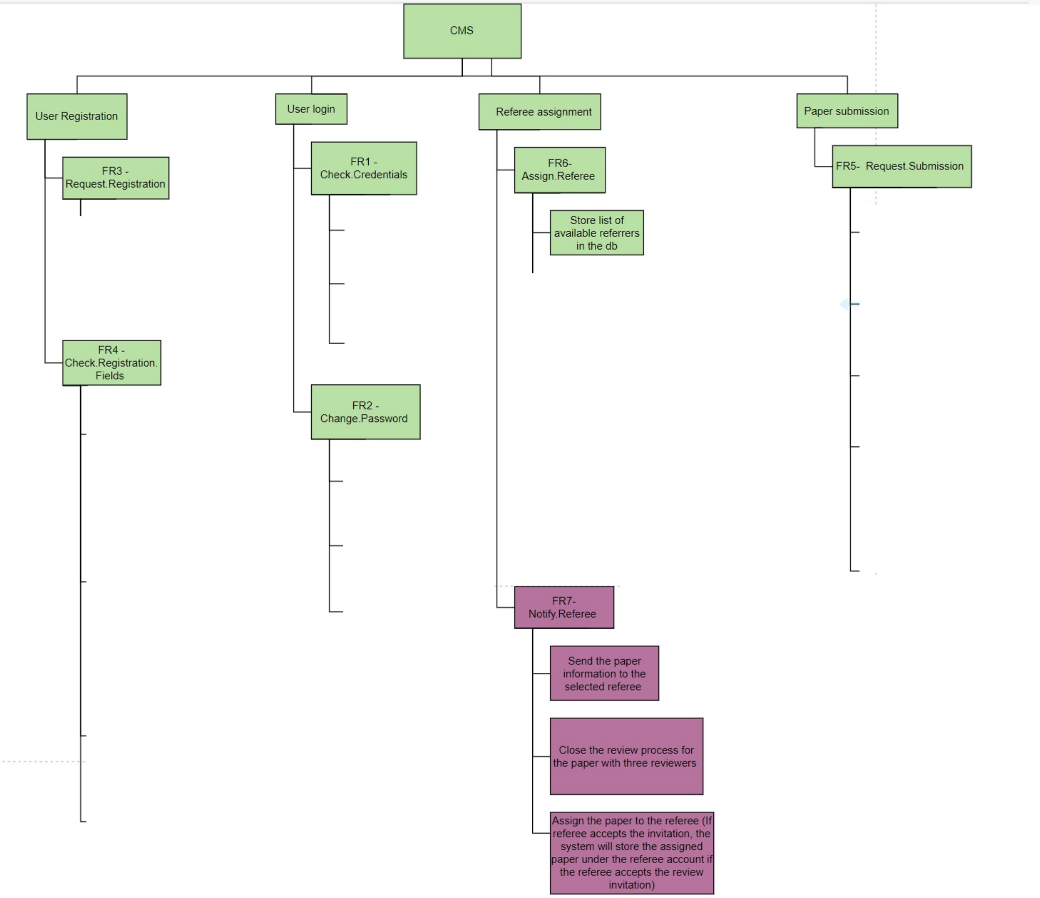
A good Example of WBS for a project.


1 Hello World System
1.1 Initiation
1.1.1 Develop Business Case
1.1.2 Assign Project Manager
1.1.3 Develop Project Charter
1.2 Planning
1.2.1 Perform Primary Planning
1.2.1.1 Develop WBS
1.2.1.2 Plan Schedule
1.2.1.3 Plan Resource
1.2.1.4 Plan Budget
1.2.2 Perform Suppl. Planning
1.2.2.2 Plan Risk Mgmt.
1.2.2.3 Plan Change and Conf.
1.3 Execution and Control
1.4.1 Manage Communications
1.4.2 Manage Procurement
1.4.3 Track and Manage Tasks
1.4.4 Execute Project Tasks
1.4.4.1 Build Subsystem A
1.4.4.2 Build Subsystem B
1.4.5 Manage Changes
1.4.6 Perform UAT
1.4 Closeout
1.5.3 Document Closeout
1.5.4 Transfer Project

https://www.workamajig.com/blog/guide-to-work-breakdown-structures-wbs

WBS

- In the previous example, all the project’s tasks were covered in the WBS.
- WBS is usually used as a project management task.
- However, in this course, we use WBS as an analyzing tool.
  - It is going to work instead of fine-grained scenarios for each functional requirement.
  - Therefore, instead of project management tasks, use extracted tasks in WBS.
  - A partial WBS for CMS can be seen in the following:


CMS
User Registration
FR3 - Request.Registration
FR4 - Check.Registration. Fields
User login
FR1 - Check.Credentials
FR2 - Change.Password
Referee assignment
FR6- Assign.Referee
Store list of available referrers in the db
FR7- Notify.Referee
Send the paper information to the selected referee
Close the review process for the paper with three reviewers
Assign the paper to the referee (If referee accepts the invitation, the system will store the assigned paper under the referee account if the referee accepts the review invitation)
Paper submission
FR5- Request.Submission

WBS

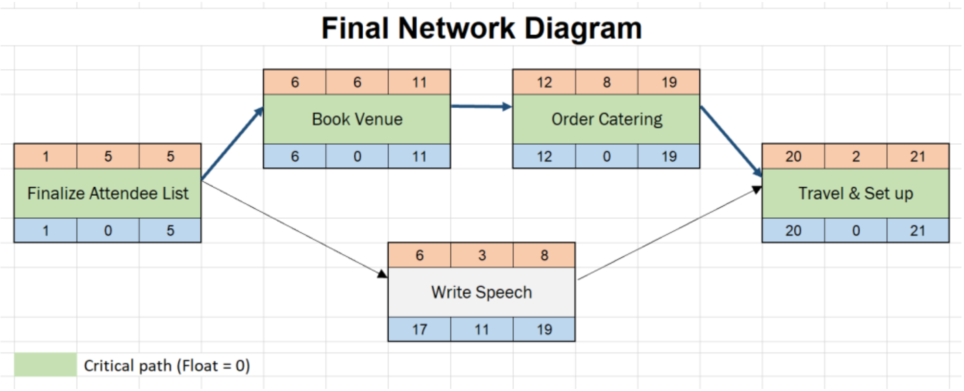
How make a WBS for this course:

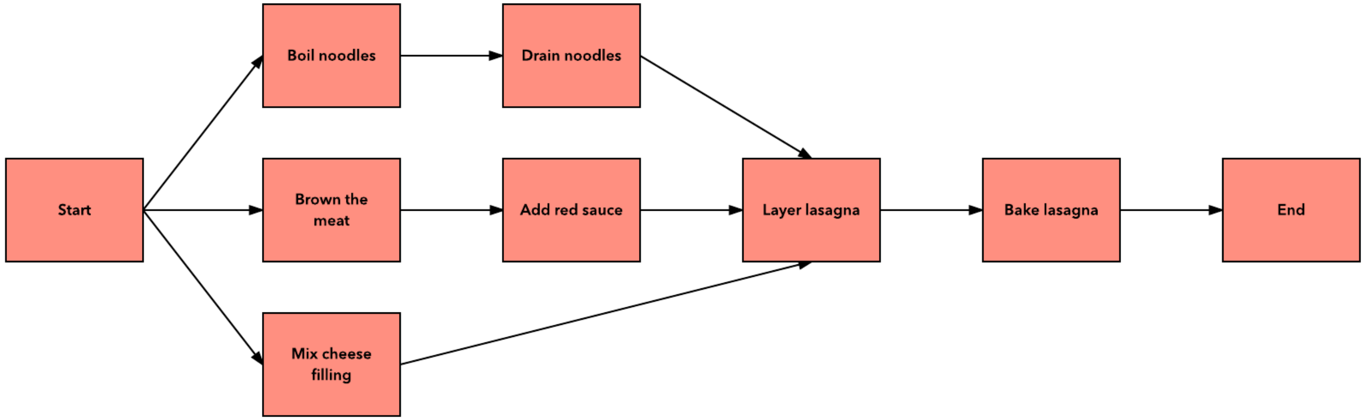
- Understand the project
- Determine functional requirements as deliverables.
- For each requirement:
  - Write (implicitly/explicitly) a scenario for the requirement
  - Extract tasks from the scenario
  - Read them and add more tasks that you should do to be able to deliver the requirement
  - (Optional) As WBS is project management tool, add all related tasks to each requirement such as planning, design, implementation, and test of each requirement (deliverable)
  - Put an ID for each task to be trackable, i.e., you can use the IDs in your project network diagram.

Marking WBS

- Check the consistency with the SRS: must include all requirements: 20%
- Logical Correctness: 10%
- UML/ notation Correctness: 5%

Network Diagram

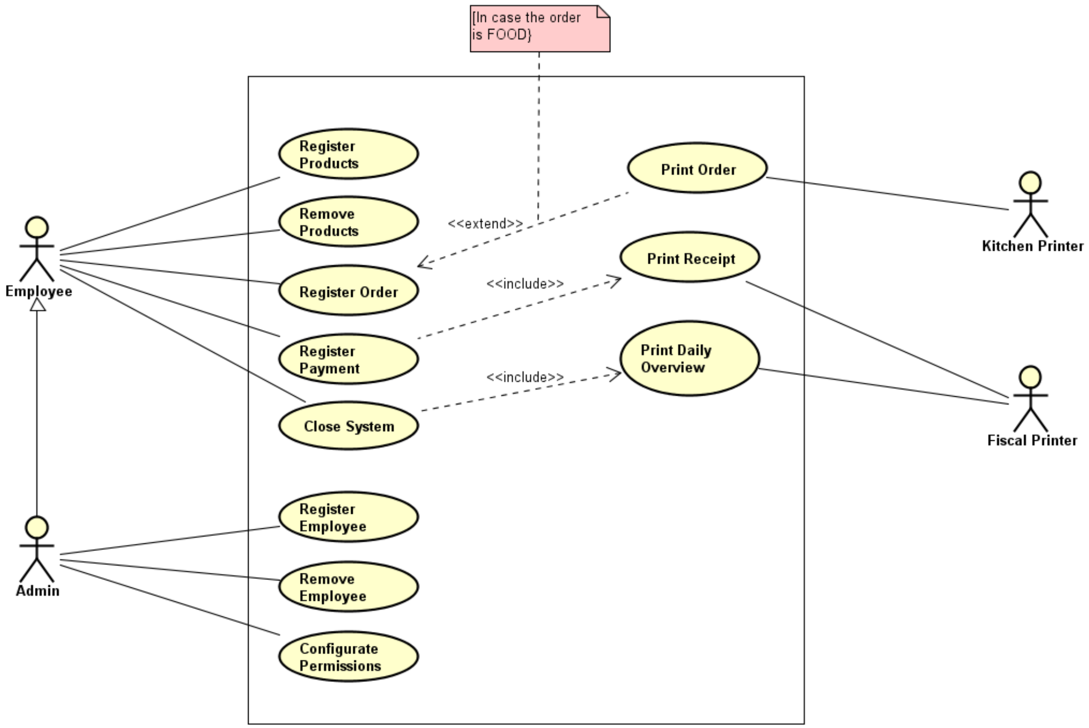


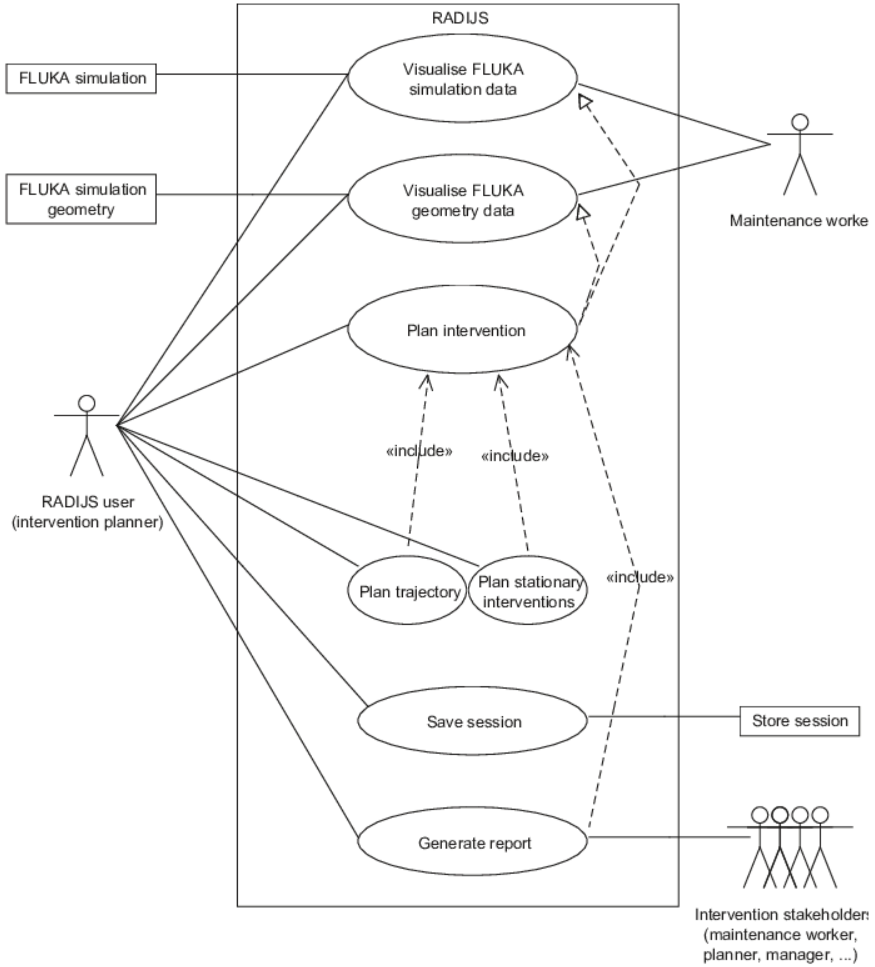


Marking Network diagram

- Check the consistency with the WBS: 10%
- Logical Correctness: 10% (planning, design, implementation, test)
- UML/ notation Correctness: 5%

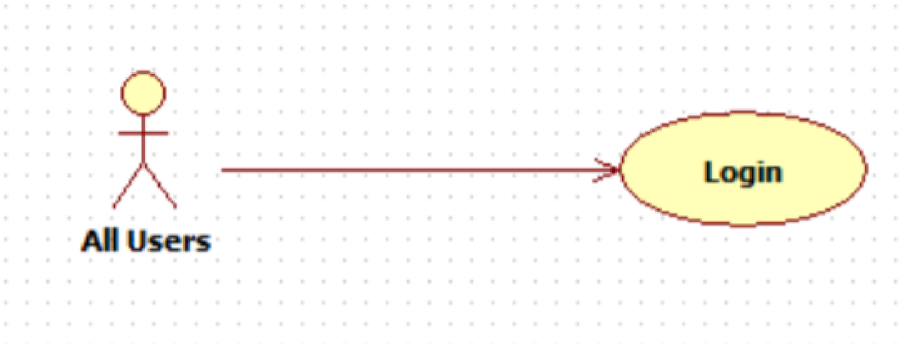
Use Case


{In case the order
is FOOD}
Register Products
Remove Products
Register Order
Register Payment
Close System
Register Employee
Remove Employee
Configurate Permissions
Print Order
Print Receipt
Print Daily Overview
<<extend>>
<<include>>
<<include>>
Employee
Admin
Kitchen Printer
Fiscal Printer


RADIJS
FLUKA simulation
FLUKA simulation geometry
Visualise FLUKA simulation data
Visualise FLUKA geometry data
Plan intervention
«include»
«include»
«include»
Plan trajectory
Plan stationary interventions
Save session
Generate report
Store session
Maintenance worke
RADIJS user
(intervention planner)
Intervention stakeholder
(maintenance worker,
planner, manager, ...)

Use Case

- Users are not necessarily same as the stakeholders
- Use cases are usually smaller than requirenets, but bigger than tasks
- Most of the time they will appear as methods in design phase



Marking:

- Check the consistency with the SRS: must include all requirements: 25%
- Logical Correctness: 10%
- UML/ notation Correctness: 5%