(1)

Page No.:
Date:
YOUVA

SRS

* Requirement - understanding b/w customer & supplier

* Specification - what the S/w must do (SRS given to technical team)
  - Req

* Req^{m} that are not in the SRS
  - cost
  - Delivery date
  - Acceptance procedures
  - } these are in BRS

* use of SRS
  - Design
  - validation
  - customer contract ra ely

* Role of SRS
  (1) SRS must be correctly define all of the SW req^{m}s, ie must be concise not lengthy
  (2) SRS should not describe design, verification or project mgmt details except req^{m} design constraints.

IEEE SRs format

6 parts

1. Introduction
   - 1.1 Purpose
   - 1.2 Document conventions
   - 1.3 Intended Audience
   - 1.4 Product scope
   - 1.5 References

1.1 → **Purpose** - specify product purpose that is going to be predicted in 2 to 4 lines.
eg. Online education system SW for undergraduate courses like BBA, BCA, BBM

1.2 - **Document convention**
describe any stds or typographical conventions used to writing this SRS.
eg fonts or highlighting have special significance

1.3 **Intended audience & Reading suggestions**
- specify users for whom this product is being developed
eg developers, project managers, marketing staff, users, testers, & documentation writers.
- Specify what the rest of this SRS contains & how it is organized.

1.4 - **Scope** - give short description of SW including -relevant benefits, objectives, & goals
eg
Relate SW to corporate goals or business strategies.
If separate vision & Scope document is available refer to it rather than duplicating its contents here.

1.5 References : List other documents & web references to which this SRS refers.
eg Interface styles guides
- contracts, standards
- system req^ms specifications
- use case documents
- vision or scope documents

To access a copy provide title, author, version, date, source of location.

(2) Overall Description

2.1 product perspective
2.2 Product functions
2.3 User classes & characteristics
2.4 operating Environment
2.5 Design & implementation constraints
2.6 user documentation
2.7 Assumptions & dependencies

2.1 → Product Perspective

- Specify whether product is a replacement of existing SRS, or a new or self-contained product.
- ie aim of product
- If it is component then relate req^ms of larger SRS to functionality of this SW. & identify interface betw them.
- Simple dia showing major components & interconnections

2.2 Product Function

- specify what all tasks SW

Here data flow dia or high level dia is effective

product is going to perform & how major components is useful

2.3 User classes & characteristics

identify users using this product into diff classes, based on frequency of use, functions being used, security, technical expertise or privilege levels, educational level of experience.

2.4 Operating Environment

Describe the environment in which SW will work, ie HW platform, OS & versions, any other SW components or appl^ns.

2.5 Design & Implementation Constraints

Specify constraints such as corporate policies, HW limitations (time, memory req^m), interfaces to other appl^ns, technologies, tools & databases to be used, Design conventions, or programming stds.

2.5 User Documentation

List user manuals, on-line help & tutorials that will be delivered along with SRS.

2.7 Assumptions & Dependencies

List any assumptions assumed factors that would affect req^ms stated in SRS
- eg third party & commercial components.
- development or operating env^ or constraints issues
- list dependencies on external factors like component from other project reuse.

(3) External Interface Requirements.

3.1 User Interfaces
Describe the logical characteristics of each interface b/w the SW product & users.
eg screen images, GUI stds

to use, screen layout constraints.
- Details of user interface design should be documented in separate user interface specification

3.2 HW interface

Describe logical & physical characteristics of each interface b/w the SW product & HW components

eg. supported device type
- data & control interaction b/w SW & HW.
- comm^n protocols to be used.

3.3 SW Interfaces.
- describe connections b/w this product and other SW like databases, OS, tools libraries etc

3.4 Communication Interfaces
- describe req^ms associated with comm^n. form eg E-mail, web browser

server comm" protocols, etc.
comm" stds used such as FTP, HTTP

(6) System features

- Major services provided all sys features specify functional req^ms for this.
- organize features by use case, mode of operation, user class, object class, functional hierarchy.

6.1 System feature 1

state feature name in few word

6.1.1. Description of Priority
Priority rated ~~on~~ benefit, penalty, cost & risk as low / to 9 high priority.

* 6.1.2. Stimulus Response sequence
list user actions & sys responses

6.1.3

6.1.7 Functional Req^m.
~~Desc~~ specify detailed functional req^ms. needed to carry

out services provided by feature.

eg respond to error conditions or invalid i/ps.

- each req^m should be given sequence no. as
  REQ 1:
  REQ-2:

6.2 System Feature 2: (~~skip~~ like 6.1)

(5) other non functional Req^m's

5.1 Performance Req^ms
specify req^m such as timing relationship for real time systems & explain them. Give specific req^ms

5.2 Safety Req^m's
specify req^ms that are concerned with loss, damage or harm that could result from use of product.

5.3 Security Req^m specify req^m regarding security & privacy.

5.4 S/w qty attributes.

specify additional quality characteristics for product.

5.5 Business rule
- specify the individuals who perform fun^n under specific circumstances.

6) Other Req^m
define any req^m s not specified in SRS.
eg db req^m s, legal req^m, reuse objectives, etc.

Appendix A : Glossary

Define all terms to properly interpret the SRS. including acronyms & abbreviations.

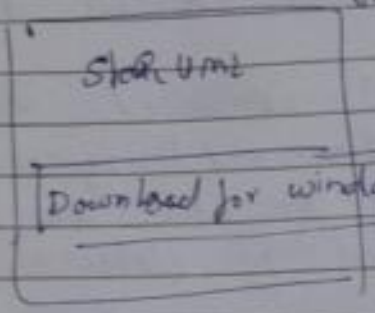
Appendix B : Analysis models.
eg data flow dia, class dia, state transition dia, ER dia

Appendix C : To be determined list

collect a numbered list of TBD (to be determined) references that remain in the SRS so they can be tracked to closure.

How to Install STARUML

1) Goto google & type [staruml]
2) Click on staruml ↵ it will take to offical website
3) .



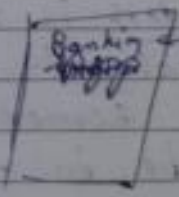
click ↵ here

4) you get staruml setup 400 . exe file downloaded of 148 MB size
5) Now click it & open
6) Then it asks for permission give yes
7) It starts installing . & after that
8) staruml gets automatically opened and Thank you for having this unregistered version box will appear.
9) Thus staruml is available now.

USE CASE diagrams

- It is methodology used in system analysis, to identify, clarify and organize system requirements.
- It shows interaction bet" systems & users in a particular environment & related to particular goal.
  It is represented using ellipse.
- use case dia has 4 diff elements
  i - systems
  ii - Actors
  iii - use cases
  iv - Relationships

(i) systems - is what you are developing it can be website, sw component, business process, an app, or any other thing.
- it is represented by rectangle [] & put name at top

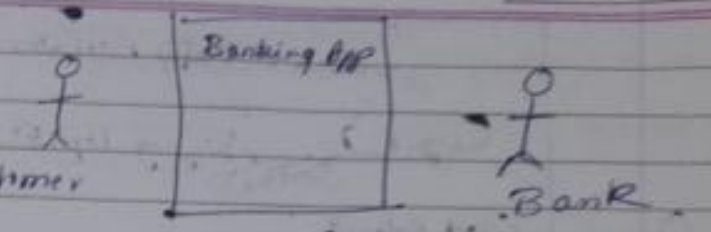


This rectangle define scope of the system. what ever happen is within this rectangle happpens within the Banking App.

ii) Actor - depicted by stick figure
- ~~It~~ is who use sys to achieve goal
- Actor can be person, another system, organization, external device.
- eg customer - use Bank Apl
- eg Bank - provide provide info that feeds into our Banking app like transactions & account balances.
- actors are external objects. & must be placed outside our systems.

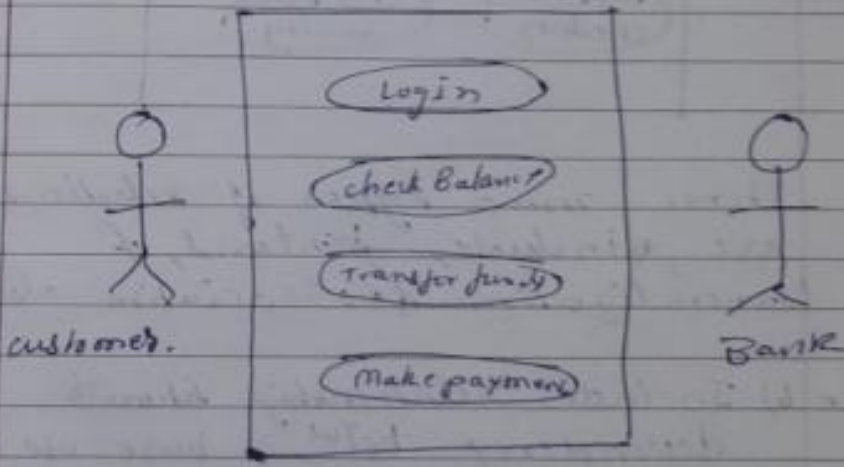
Primary Actor initiates the use of the system eg customer. It should be to left of the system

Secondary Actors is reactionary. Eg Bank is going to act once customer does something. It should be to right of the sys

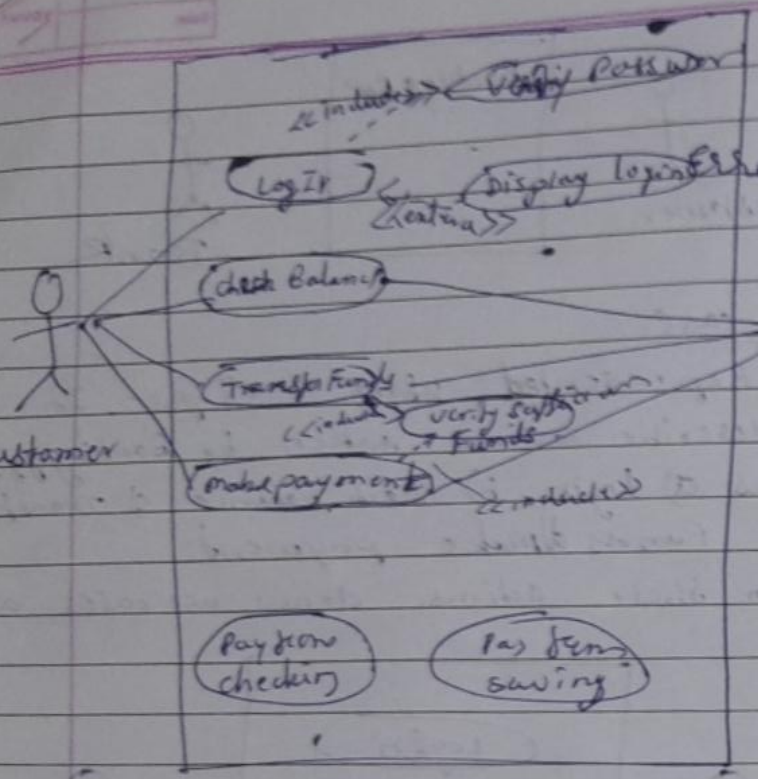


iii) Use case
- Represented using ellipse. It describe actions taken by Banking app as a) Login b) check balance c) Transfer Funds d) Make payment.

For these actions draw use cases as



iv) Relationships : draw solid line bet^w actor & use case to show relationship. called the relationships called association, signifies a basic communication or interaction



Three more types of relationship are a) include, b) extend, c) Generalization & d) association also.

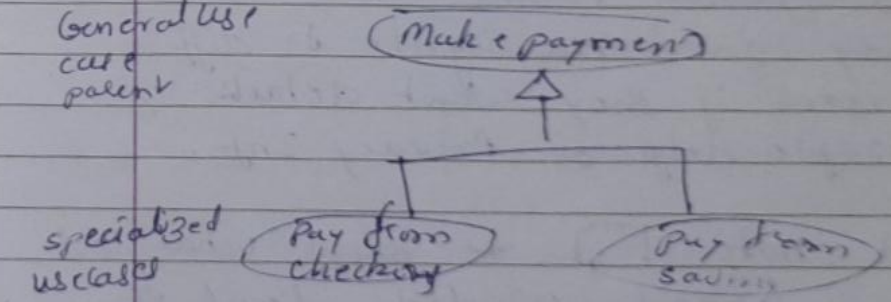
- b) Include relationship shows dependency b/w a base use case & included use case.
  - If base use case executed then include usecase is executed
  - shown with --> from base usecase to include usecase
  - base is complete if include usecase is executed

ie eg (Log In) - - - -> (Verify Password)
base usecase — Include usecase

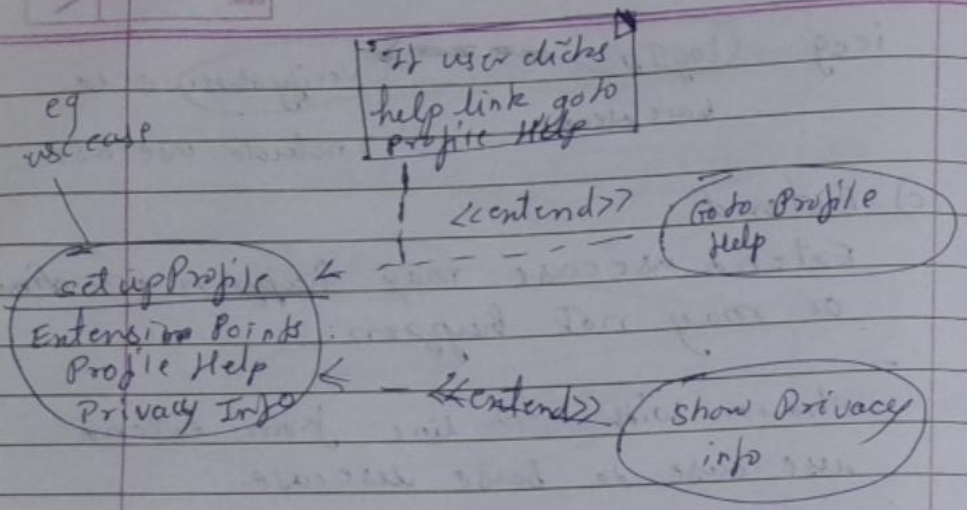
c) Extend:
Extend usecase may happen sometime or may not happen.
- shown with --> line from extend use case to base usecase.

(Log In) <- - - - - (Display login) errors

d) Generalization



- we can have Generalization for actors
- Extension points: are detailed version of extend relationships.



This use case shows customer can set up their profile in Banking app.

Extension Points tells that they can navigate to diff screens if they want details of Profile Help or Privacy info.

Add note ☐ to show what sort of conditions would lead to these extension points

Jenkins

- open source automation tool
- written in Java
- allows continuous integration.

- it builds & test sw proj continuously making it easier for developers to integrate changes to proj.
- &
  make easier for user to obtain a fresh build.

- allows continuous delivery of sw by integrating testing & development technologies

- support CI/CD using pipeline. & automates other routine development activities. tasks

- development process is fast through automation.

- acheives CI with help of plugins
- plugins allows integration of various DevOps stages.