

Assignment 1

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

1. Ainan Ahmed Chowdhury--1377020
2. Shrabanti Saha Rimi--1377509
3. Ashis Banik -- 1377253
4. Aktari Sadia Sabah--1382413
5. Syed Fawzul Azim--1364224

Question 1:

Please give examples for

- Safety Critical Systems (SCS)
- Safety Related Systems
- Mission Critical Systems
- Business Critical Systems

Please explain, why systems fall into a certain category and what the characteristics are!

Safety critical systems (SCS)

Safety critical systems (SCS) are systems designed with the intent of curbing the effects of an accident from a hazardous event.

There are many well-known examples in application areas such as the aviation industry, the medical profession, nuclear testing, financial sector, aircraft flight control, weapons and nuclear systems.

It is an application where human safety depends on the correct usage of the software program. The software or the hardware must not contribute to the cause of the accident or escalate the accident, which is usually unsafe.

Safety critical systems are heavily dependent on computers, so it is up to these computers to ensure that no failure occurs in the usage of these systems, a failure in such a system could trigger abnormal directional movements. The most valued property of the system is that it is dependable and dependability shows the users trust in that system. The dependability of the system hinges on the ability of the system to deliver services when required, as the services are specified, and the ability of the system to protect itself from intrusion[1]

Safety critical systems make use of electrical programming technologies which interact with mechanical systems and a human interface for interaction.[1]

Since in a SCS development process software development is strictly intertwined with system development, many of the used techniques have been derived directly from system level techniques; but tailoring them to software is not so immediate and has not always produced the expected results. As a matter of fact, implementing such techniques for software often requires very costly solutions to achieve adequate performance.[2]

Safety Related Systems

A safety related system is a system whose failure could directly or indirectly cause safety risks for the people or environment involved.

A system can be characterized as a safety related system if it provides functions which considerably reduce the risk of a hazard, and in combination with other risk reduction measures, reduces the overall risk to a tolerable level, or if it is required to function to maintain or achieve a safe state for the equipment under control. [3]

Safety-related systems are designed to implement safety functions in order to achieve or maintain safe states of equipment/ system/ installation, in respect to specific hazardous events. In this context, functional safety is the part of the overall safety relating to equipment/ system/ installation and their control systems that depends on the correct functioning of the safety-related systems.[4] These functions are known as the safety functions of the system or device and are the ability to prevent initiation of a hazard or detect the onset of a hazard, and to take the necessary actions to terminate the hazardous event, achieve a safe state, or mitigate the consequences of a hazard.

All elements of the system that are required to perform the safety function, including utilities, are safety related, and should be considered part of the safety-related system (SRS). [3]

A typical example of safety related systems could be Database Systems.. A mismanagement of data is a significant risk of a hazard. So, every database system is built with safety keeping in mind. By ensuring safety of the data, systems safety can be ensured.

Another example of safety related systems is the Train Protection System. It is a system which ensures safety in the event of human-caused errors. In such a system, various safety functions are implemented. A common safety function of this system is an automatic warning system which provides warnings and visual reminders to train drivers of approaching signals, reduction in speed etc. Another common safety function of train protection systems is automatic brake application. Which only occurs if the driver fails to acknowledge a warning.

Mission Critical Systems:

Among those four critical systems, the mission-critical system plays a vital role. A mission critical system is a system that is very much important to the existence of a business or organization. This system is one that, whether it fails or is disrupted, will bring an entire operation or organization to a halt. It is the best part of a business if the authority wants to protect their data and application successfully. Therefore a mission-critical system is also called mission essential equipment or mission critical application.[5]

IT Business, online banking system, electric power systems, water treatment facility pumps, a navigational system for a spacecraft, railway/aircraft operating and control systems and so many computer systems that will negatively affect business and society when they fail are examples of mission-critical systems.

If we talk about electric power systems or water treatment facility pumps, when an electric power system fails or water treatment facility pumps are stopped, all other works related to it are forced to stop. Since these processes are disrupted, businesses suffer significant consequences and hampered productivity. On the other hand, in IT businesses and organizations, whatever it's a cyberattack, a power failure, or defective hardware, the management team always wants to ensure the smooth flow of work outcomes. So, the database server, database system, and process need to be protected from any kind of incident that can be a cause of huge loss of data. Database systems and process control servers are mission-critical systems.

A system may be considered mission critical when any of the following conditions are met:

If Human life or wellbeing is put at risk, Data or research has been compromised, Severe damage is done to one's reputation, Critical corporate functions and applications have been disrupted, if there is a loss of data or access to data is experienced.[6]

Business Critical Systems:

Basically, when strategic frameworks are important for the achievement of a business, they become business-critical systems. At the point when a business basic framework comes up short or is interfered with, associations can confront monetary misfortunes, client disappointment, and decreases inefficiency.[7]

Business-critical systems are customized to stay away from critical substantial or immaterial monetary expenses; e.g., loss of business or harm to notoriety. This is often due to the interruption of service caused by the system being unusable. Examples of a business-critical systems:

- 1.The customer accounting system in a bank,
- 2.stock-trading system,
- 3.ERP systems of a company,
- 4.Internet search engine.
- 5.Online banking systems.
- 6.Cloud-based data storage and networking systems.

The business criticality is directed by the run-of-the-mill conveyed climate and the estimation of information utilized by the application. Factors that determine business criticality are: reputation damage, financial loss, operational risk, sensitive information disclosure, personal safety, and legal violations.[8]

Adding constant correspondence into business-critical incorporated arrangements and new smart gadgets is troublesome with a large number of cloud administrations available today. While current traditional interchange stages don't meet the assorted arrangement of necessities for business- or safety-critical solutions, [iotcomms.io](https://www.iotcomms.io) is built specifically for such use-cases.[9]

1. Reliability
- 2.Quality
- 3.Longevity and stability
- 4.Scalability

Reference

1. <https://www.technipages.com/definition/safety-critical-system>
2. Pietrantuono, Roberto & Russo, Stefano. (2013). Introduction to Safety Critical Systems. 10.1007/978-88-470-2772-5_2.
3. [Plant Safety & SIS - Safety Related Systems \(SRS\)](#)
4. Florent Brissaud, Didier Turcinovic. Functional Safety for Safety-Related Systems: 10 Common Mistakes. 25th European Safety and Reliability Conference, Sep 2015, Zurich, Switzerland. fhal01199081f
5. [Your Data: How to Decide What's Mission-Critical](#)
6. <https://www.gbtech.net/mission-critical-systems-and-why-you-need-them-managed/>
7. <https://www.netmotionsoftware.com/blog/mobility/mission-critical-systems>
8. https://help.veracode.com/r/review_assurancelevels
9. <https://iotcomms.io/characteristics-of-a-cpaas-for-business-critical-solutions-part-1/>

Assignment 2

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

1. Ainan Ahmed Chowdhury--1377020
2. Shrabanti Saha Rimi--1377509
3. Ashis Banik -- 1377253
4. Aktari Sadia Sabah--1382413
5. Syed Fawzul Azim--1364224

Question 1:

Emergence:

Emergence refers to the existence or formation of collective behaviors - what parts of a system do together that they would not do alone.

In describing collective behaviors, emergence refers to how collective properties arise from the properties of parts, how behavior at a larger scale arises from the detailed structure, behavior and relationships at a finer scale. For example, cells that make up a muscle display the emergent property of working together to produce the muscle's overall structure and movement. A water molecule has emergent properties that arise out of the properties of oxygen and hydrogen atoms. Many water molecules together form river flows and ocean waves. Trees, other plants and animals form a forest.

The emergence of order and organization in systems composed of many autonomous entities or agents is a very fundamental process. The process of emergence deals with the fundamental question: "how does an entity come into existence?" In a process of emergence, we observe something (for instance the appearance of order or organization) and ask how this is possible, since we assume causality: every effect should have a cause. The surprising aspect in a process of emergence is the observation of an effect without an apparent cause. Although the process of emergence might look mysterious, there is nothing mystical, magical or unscientific about it.[1]

If we consider the world of emergent properties, the deepest mysteries are as close as the nearest seedling, ice cube, grain of salt or pile of sand, as Laughlin explains in his book [Laughlin05]. It is doubtful that the ultimate laws can be found at inconceivable high energies or extreme scales, if we do not understand things at our own scale well enough. In other words, we must step back and look at the patterns and the interactions of everyday objects to discover the nature of our universe.[1]

Emergent phenomena:

emergent phenomena are those for which the amount of computation necessary for prediction from an optimal set of rules, classifications and analysis, even derived from an idealized perfect understanding, can never improve upon the amount of computation necessary to simulate the system directly from our knowledge of the rules of its interactions.[2]

The following examples of emergent systems demonstrate the kinds of feedback between individual elements of natural systems that can give rise to surprising ordered behavior. They also illustrate a clear trade-off between the number of elements involved in the emergent system and the complexity of their individual interactions. The more complex the interactions between elements, the fewer elements are needed for a higher-level phenomenon to emerge. Hurricanes

and sand dunes form from vast numbers of very simple elements whereas even small groups of birds can exhibit flocking behavior.

Flocks of birds and hurricanes are real-time dynamic phenomena with no lasting structure. But much emergent behavior is due to persistent changes to the local environment. Sand dunes, termite mounds, and cities are persistent physical structures that organize the behavior of the very entities that build them. That is, emergence often gives rise to *stigmergy* structures. Stigmergy structures emerge in the Internet too, based on persistent structures such as databases, wikis, blogs, and the Web as a whole.

Question 2:

Please write code to plot the Mandelbrot set!

Explore and play with fractals!

Mandelbrot Set: The mandelbrot set is defined by the set of complex numbers C for which the complex numbers of the sequence Z_n remain bounded in absolute value. The Mandelbrot Set (M-Set in short) is a fractal. It is plotted on the complex plane.

The sequence Z_n is defined by:

$$Z_0 = 0$$

$$Z_{n+1} = Z_n^2 + C$$

Here, both Z and C are complex numbers. C is the complex number for which we are testing whether or not it's in the M-Set.

So, how do we calculate the Mandelbrot set? For a complex number C , we first set $Z = 0$, then set Z to $Z^2 + C$. If we do this over and over again, there are 2 possibilities for Z - either it escapes to infinity or it does not. If it does not escape to infinity, it's in the mandelbrot set.

Now, how do we calculate whether it will escape to infinity? As we know, for every value of C in the mandelbrot set, the sequence Z_n has to be bounded in absolute value. We assume the sequence is not bounded if the modulus of Z_n in any iteration is greater than 2. Also, we will iterate the sequence Z_n 40 times for a complex number C to see whether Z_n is greater than 2 at any iteration. If the Z_n does not exceed 2 after 40 iterations, we can say that the complex number C is in the mandelbrot set.

Code:

For coding to plot the mandelbrot set, we used python as the programming language, numpy as data structure and matplotlib for plotting as visualization.

1. First, we import our necessary data structure and library. Also we defined our maximum iteration limit as a global variable.

```
import numpy as np
import matplotlib.pyplot as plt
MAX_ITER = 50
```

2. Then, we define the mandelbrot function. Here, it will return the number of iterations needed to reach modulus of $Z_n > 2$. If it doesn't exceed 2 within MAX_ITER iterations, we simply return MAX_ITER.

```
def mandelbrot(c):  
    z = complex(0,0)  
    for i in range(1, MAX_ITER):  
        z = z**2 + c  
  
        if(abs(z) > 2):  
            return i  
    return MAX_ITER
```

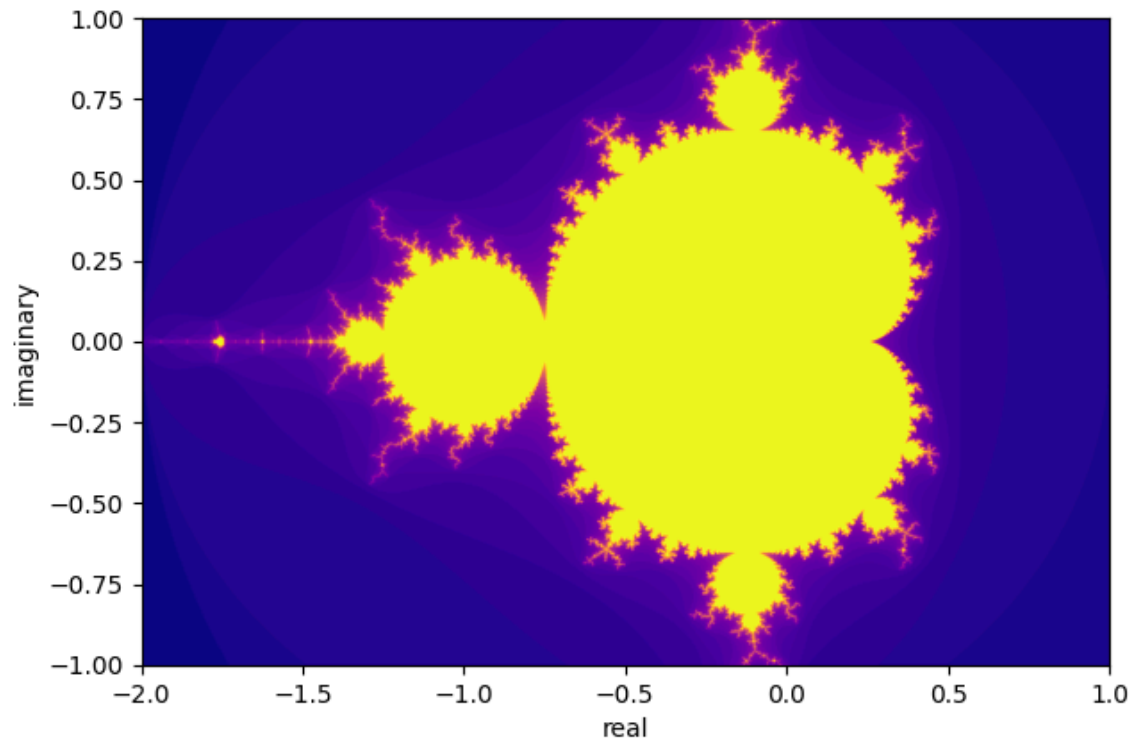
3. Now, comes the main function. First, we get 2 arrays X, Y as real and imaginary numbers. We used the linspace function from numpy to get a list of values within a certain limit. We also defined a Matrix Z to store our result. Then, for every value in X and Y we called the mandelbrot function and stored the returned result in Z.

```
X = np.linspace(-2,1,2000)  
Y = np.linspace(-1,1,2000)  
Z = np.zeros((2000,2000))  
for ix,x in enumerate(X):  
    for iy,y in enumerate(Y):  
        Z[ix,iy] = mandelbrot(complex(x,y))
```

4. Finally we plot the result using matplotlib.

```
plt.figure(figsize=(7, 12), dpi=100)  
plt.imshow(Z.T, cmap='plasma', extent=[-2,1,-1,1])  
plt.xlabel("real")  
plt.ylabel("imaginary")  
plt.show()
```

Plot:



Question 3:

(a) Software measurement has been recognized as a crucial component of successful software engineering practice. A software measurement is a titrate impute of a feature of a software product or procedure. Characteristics of a good measurement are reliability, validity, sensitivity.[3] On the other hand, making quantitative/qualitative decisions besides risk assessment and reduction in software projects we need software metrics.[3] Software metrics are important for a variety of purposes, including measuring software efficiency, planning work items, calculating output, and a variety of other applications.

(b) Categories of metrics: From a measurement perspective, software metrics can be divided into three parts.

- Product metrics
- Process metrics
- project metrics

Product Metrics: Product metrics describe the characteristics of the product. Product metrics can be size metrics, the complexity metrics, internal or external attribute measurement of the products.[4]

Process Metric: To improve software development and maintenance process metric can be used. It mainly helps to measure the parameters of software development life cycles like duration estimate, cost assessment, effort required, process quality and effectiveness/efficiency of the development process.[4]

Project Metric: Project metric describes the features of the project as well as its implementation.

(d) Cyclomatic Complexity: Three plan standards are noticeable in programming improvement exemplification, information stowing away, and division of concerns. These standards are utilized as emotional quality models for both procedural and item arranged applications. The reason for research is to measure exemplification, information covering up, and division of concerns is evaluated utilizing cyclomatic-based measurements. Because of this exploration, the inferred plan measurements, coefficient of exemplification, coefficient of information stowing away, and coefficient of the detachment of concerns, are characterized and applied to creation programming demonstrating whether the product has low or high embodiment, information covering up, and partition of concerns. [5]

(e) Cyclomatic Complexity used: The motivation behind this archive is to portray the organized testing philosophy for programming testing, otherwise called premise way testing. In light of the cyclomatic intricacy proportion of McCabe, organized testing utilizes the control stream construction of programming to build up way inclusion rules. The resultant test sets give more exhaustive testing than explanation and branch inclusion. Augmentations of the crucial organized testing strategies for mix testing furthermore, object-situated frameworks are additionally introduced. A few related programming intricacy measurements are portrayed. Outlines of specialized papers, contextual analyses, and observational outcomes are introduced in the informative supplements.

In light of the cyclomatic intricacy proportion of McCabe, organized testing utilizes the control stream design of programming to set up way inclusion rules. The resultant test sets give more exhaustive testing than proclamation and branch inclusion. Augmentations of the principal organized testing strategies for coordination testing and article arranged frameworks are additionally introduced. A few related programming intricacy measurements are portrayed. Synopses of specialized papers, contextual investigations, and observational outcomes are introduced in the indices.[6]

Question 4:

(a)

Reliability:

Reliability is the probability of failure-free operation of a system over a specified time within a specified environment for a specified purpose. In other words, Reliability is a measure of how closely a system matches its stated specification.

Software Reliability:

- It is difficult to define the term objectively.
- Difficult to measure user expectations,
- Difficult to measure environmental factors.
- It's not enough to consider simple failure rate:
 - Not all failures are created equal; some have much more serious consequences.
- Might be able to recover from some failures reasonably.

Failures and Faults:

- A failure corresponds to unexpected runtime behavior observed by a user of the software.
- A fault is a static software characteristic which causes a failure to occur.
- Not every fault causes a failure:
 - Code that is “mostly” correct.
 - Dead or infrequently-used code.
 - Faults that depend on a set of circumstances to occur.

Improving Reliability:

Primary objective: Remove faults with the most serious consequences.

Secondary objective: Remove faults that are encountered most often by users.

Reliability Metrics:

- Probability of Failure on Demand (POFOD):
 - Likelihood that system will fail when a request is made.

- POFOD of 0.001 means that 1 in 1000 requests may result in failure.
- Any failure is important; doesn't matter how many if > 0
- Relevant for safety-critical systems
- Rate Of Occurrence Of Failure (ROCOF):
 - Frequency of occurrence of failures.
 - ROCOF of 0.02 means 2 failures are likely in each 100 time units. •
- Relevant for transaction processing systems.

The Cost of Reliability:

- In general, reliable systems take the slow, steady route:
 - trusted implementation techniques
 - few uses of short-cuts, sneak paths, tricks
 - use of redundancy, run-time checks, type-safe pointers
- Users value reliability highly.
- “It is easier to make a correct program efficient than to make an efficient program correct.”
- Cost of software failure often far outstrips the cost of the original system:
 - data loss
 - down-time
 - cost to fix

Measuring Reliability:

- Hardware failures are almost always physical failures (i.e., the design is correct).
- Software failures, on the other hand, are due to design faults.
- Hardware reliability metrics are not always appropriate to measure software reliability but that is how they have evolved.

(b)

Probability of faulty components, $P(F) = 4\%$ or 0.04

So, Probability of not faulty components, $P(F') = 100 - 0.04 = 0.96 = 96\%$

90% probability that a faulty components is detected, $P(D|F) = 90\% = 0.9$

2% of the time un-faulty components are recorded as faulty, $P(D|F') = 2\% = 0.02$

Applying Bayes' Formula,

The probability of detected faulty component being actually faulty is

$$P(F|D) = P(D|F) \times P(F) / P(D)$$

Here, $P(D)$ is the probability of detection which is a combination of two dis-joints events

$$P = P(D \cap F) + P(D \cap F')$$

$$P(D) = p(D|F) \times P(F) + p(D|F') \times p(F') = (0.9 \times 0.04) + (0.02 \times 0.96) = 0.0552$$

$$\text{Therefore, } P(F|D) = P(D|F) \times P(F) / P(D) = 0.9 \times 0.04 / 0.0552 = 0.652 = 65.2\%$$

Reference:

1. Types and Forms of Emergence Jochen Fromm Distributed Systems Group, Electrical Engineering & Computer Science, Universität Kassel, Germany
2. Emergent Phenomena and Complexity Vince Darley Division of Applied Sciences Harvard University 33 Oxford Street, Cambridge MA 02138
3. SOFTWARE MEASUREMENTS AND METRICS: ROLE IN EFFECTIVE SOFTWARE TESTING, Sheikh Umar Farooq*Research scholar, P.G. Department of Computer Sciences, University of Kashmir Srinagar, J&K – 190006.S. M. K. Quadri Head, P.G. Department of Computer Sciences, University of Kashmir, Srinagar, J&K – 190006.Nesar Ahmad, University Department of Statistics and Computer Applications, T. M. Bhagalpur University, Bhagalpur-812007.
4. ANALYSIS OF SOFTWARE QUALITY USING SOFTWARE METRICS, Dept. of Software Engineering and Computer Science, Wolkite University, Ethiopia 2Dept. of Service and Information System Engineering, Polytechnic University of Catalonia, Spain
5. Cyclomatic Complexity-Based Encapsulation, Data Hiding, and Separation of Concerns CW Butler, TJ McCabe - Journal of Software Engineering and Applications, 2021 - scirp.org
6. NIST Special Publication 500-235, Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric, Arthur H. Watson, Thomas J. McCabe

[Thomas J. Watson Research Center](#)

Assignment 3

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

- 1.Ainan Ahmed Chowdhury--1377020
- 2.Shrabanti Saha Rimi--1377509
- 3.Ashis Banik -- 1377253
- 4.Aktari Sadia Sabah--1382413
5. Syed Fawzul Azim--1364224

Question-1

(a) Please browse the literature, e.g. IEEE Xplore or the ACM libs, and review the field of Safety Critical Systems (SCS) in Medicine! Please quote the publications correctly!

I have reviewed a few publications and research papers and I found a paper about safety critical systems in medicine from 'Academia'. Paper name is 'SAFETY CRITICAL SYSTEMS IN MEDICINE' and written by M.SPANDAN and J.VARUN CHANDRA.[1]

(b)Please summarize at least one publication!

A safety critical system is one that must function correctly to avoid human injury, human death, damage to property, financial loss, damage to the natural environment. Aircraft, cars, weapons systems, medical devices, and nuclear power plants are the traditional examples of safety-critical software systems.

Nowadays, many medical facilities are actually distributed computer systems, such as i.Heart-Lung machine. ii.Mechanical ventilation machines. iii.Radiation Therapy machines. iv.Robotic surgery machines and this paper focused on Therac-25(Radiation Therapy machine). The summary of the analysis appeals for 10 changes to Therac-25 hardware;the most significant of these are interlocks to backup software control of both electron scanning and beam energy selection. The changes recommended have several distinct objectives: improve the protection it provides against hardware failures; provide additional reliability via cross-checking; and provide a more maintainable source package. The software code for Beam Shut-Off, Symmetry Control, and Dose Calibration was found to be straight-forward and no execution path could be found which would cause them to perform incorrectly. Inspection of the Scanning and Energy Selection functions, which are under software control; however, software inspection was unable to provide a high level of confidence in their reliability. This was due to the complex nature of the code, the extensive use of variables, and the time limitations of the inspection process.[1]

(c) Define different categories of SCS in Medicine!

- **Heart-lung machines:** Cardiopulmonary bypass (CPB) is a technique in which a machine temporarily takes over the function of the heart and lungs during surgery, maintaining the circulation of blood and the oxygen content of the patient's body.
- **Mechanical ventilation systems:** Mechanical ventilation, assisted ventilation or intermittent mandatory ventilation (IMV), is the medical term for artificial ventilation where mechanical means are used to assist or replace spontaneous breathing.

- **Radiation therapy machines:** Radiation therapy or radiotherapy, often abbreviated RT, RTx, or XRT, is a therapy using ionizing radiation, generally provided as part of cancer treatment to control or kill malignant cells and normally delivered by a linear accelerator. Radiation therapy may be curative in a number of types of cancer if they are localized to one area of the body.
- **Robotic surgery machines:** Robotic surgery are types of surgical procedures that are done using robotic systems. Robotically-assisted surgery was developed to try to overcome the limitations of pre-existing minimally-invasive surgical procedures and to enhance the capabilities of surgeons performing open surgery.
- **Dialysis machines:** In medicine, dialysis (from Greek διάλυσις, dialysis, "dissolution"; from διά, dia, "through", and λύσις, lysis, "loosening or splitting") is the process of removing excess water, solutes, and toxins from the blood in people whose kidneys can no longer perform these functions naturally. This is referred to as renal replacement therapy.

(d) Which scientific disciplines have to work together to build these systems?

Software engineering for safety-critical software basic programming is for the most part hard. Specialists should suffer a heart attack comprehension of the product's part in, and connections with, the framework. Designing exercises must consent to exceptionally directed global norms. These norms regularly make the turn of events measure unbending, unfit to oblige changes, causing late mix and expanding the expense of improvement. Current utilitarian security guidelines frequently portray programming advancement as severe successive interaction with particular stages for requirements, architecture, plan, coding and comparing testing at expanding levels eventually.[2]

1. Acceptance Test Driven Development
2. Test Driven Development
3. Test-Driven Approach for Safety-Critical Software Development
 - 3.1. Classification of Requirements
 - 3.2. Writing Acceptance Test Cases
 - 3.3. Test-Driven Development and Test-Driven Design
4. Research Method
5. Result and Analysis

(e) Which Computer Science fields are important?

The major subfields of computer science incorporate the customary investigation of PC design, programming languages, and software development. In any case, they additionally incorporate computational science (the utilization of algorithmic procedures for demonstrating logical information), designs and perception, human-PC connection, data sets and data frameworks, organizations, and the social and expert issues that are special to the act of computer science.[3]

Experimental computer science is best on issues that require complex programming arrangements like the production of programming improvement conditions, the association of information that isn't even, or the development of devices to tackle obliged streamlining issues. The methodology is to a great extent to distinguish ideas that work with answers for an issue and afterward assess the arrangements through the development of model frameworks. The investigation in various fields (search, programmed hypothesis proving, planning, NP-complete issues, characteristic language, vision, games, neural nets/connectionism, AI) is additionally utilized in CS and is depicted by approach.[4]

1(f) Which new developments will foster medical SCS?

At the moment, computer technology plays a significant role in the medical industry. Heart-Lung machine, Mechanical ventilation machines, Radiation Therapy machines, Robotic surgery machines, Defibrillator machines, Dialysis machines, 3D printing, tumors model, 3D Printed Casts on a Broken Limb, Tissue Engineering With the Aid of Printers, Telemedicine, E-Healthcare these are the new development in medical science.

(g) What are challenges for the design of medical devices?

- Thermal management:
- Material selection:
- Mechanical endurance:
- Protection against electrical hazard,
- Industrial and mechanics design solutions,
- Failure mode and effects, Life cycle management

(h) What could be requirements by the users, i.e. patients, medical personnel?

Measuring and meeting user expectations during the production of medical devices would result in good products that increase patient safety, system efficacy, and eliminate product recalls and modifications. Designing for patient safety and reduced human error is an extremely important requirement. Satisfying user requirements in medical devices should also cover usability such as comfort, effectiveness, ease of both use and learning, training, hygiene requirements, maintainability and servicing, storage, labelling and so on. Attention to comfort, aesthetics and portability can affect patient readiness to follow a treatment regime, particularly for self-administered devices.

Question-2

(a)

Codes of ethics are developed to guide the behavior of members in their respective societies and professional associations. Obviously, codes are just one method that professionals can use in making judgments [***].

Ieee Code of Ethics

The following is the official IEEE Code of Ethics:

1. to accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment.
2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist.
3. to be honest and realistic in stating claims or estimates based on available data;
4. to reject bribery in all its forms.
5. to improve the understanding of technology, its appropriate application, and potential consequences.

(b)CMM - Capability Maturity Model:

It focuses on elements of essential practices and processes from various bodies of knowledge. CMM is a method to evaluate and measure the maturity of the software development process of an organization. CMM measures the maturity of the software development process on a scale of 1 to 5. CMM v1.0 was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University in Pittsburgh, USA. CMM helps to solve the maturity problem by defining a set of practices and providing a general framework for improving them. The focus of CMM is on identifying key process areas and the exemplary practices that may comprise a disciplined software process.

Staged Representation:

The staged representation is the approach used in the Software CMM. It is an approach that uses predefined sets of process areas to define an improvement path for an organization. This improvement path is described by a model component called a Maturity Level. A maturity level is a well-defined evolutionary plateau towards achieving improved organizational processes.

Continuous Representation:

Continuous representation is the approach used in the SECM and the IPD-CMM. This approach allows an organization to select a specific process area and make improvements based on it. The continuous representation uses Capability Levels to characterize improvement relative to an individual process area.

+Safe

+SAFE, V1.2 is an extension to the continuous representation of CMMI® for Development, Version1.2 (CMMI-DEV, V1.2) [SEI 2006]. This extension consists of two process areas added to CMMI-DEV to provide an explicit and focused basis for appraising or improving an organization's capabilities for providing safety-critical products. A key aim of +SAFE is to identify the safety strengths and weaknesses of product and service suppliers, and to address identified weaknesses early in the acquisition process. The safety extension was developed so

that CMMI appraisers and users can become familiar with the structure, style, and informative content provided to reduce dependence on safety domain expertise. This extension was developed for standalone use. It is not intended to be embedded in a CMMI model and it does not rely on any specific safety standards. There are intentional overlaps with CMMI model content and with some safety standards.

(c)

V-Model-XT

Basic Ideas:

The V-Modell is the German system development and lifecycle process model standard for federal administration and defense engineering projects. The V-Model regulates the system development and maintenance process; it defines binding sets of activities and artifacts, and accompanying processes such as quality assurance, configuration management, and technical project management. It is publicly available and many companies have successfully adopted it.[4]

In 2002, the most recent update of the V-Modell dated back to 1997. Many new innovations in software engineering were missing, as were quality attributes of the process model. At the end of 2004 the new V-Modell XT was established within the federal administration, military engineering projects, and companies as the new development process standard [5]. The addition XT means 'extended Tailoring' and underlines the flexible adaptability to specific project requirements.

Key parts of the model

This model has a complex structure of components, including Project Types, Process Modules, Project Execution Strategies. These components are connected with others and with the 'extended tailoring' provide a new methodical way to develop the system and project.

- **Project Type**

Based on two main characteristics of the project which are Subject of the project and Project Role, there exists 4 product types definition in the V-Model XT:

- System Development Project of an Acquirer.
- System Development Project of a Supplier.
- System Development Project of a Supplier with the Acquirer.
- Introduction and Maintenance of an Organization – Specific Process Model.

- **Process Modules**

a process module encapsulates a set of products, activities, and roles. Products represent the What of a project, meaning all (provisional) results, like documents, source codes or physical components.

A process module contains a specific set of products, activities, and roles relevant to a specific process area. All contents of a process module have content dependencies on each other.[5]

- **Project Execution Strategies**

A project's execution is usually very complex. To enable reliable planning and controlling of a project, an ordered process has to be worked out. To support

the users, the V-Modell XT includes a catalog of so-called strategies for project operations (SPO). SPOs contain the dynamic parts of the V-Modell XT. [5]

Suitability for designing SCS.

A safety-critical computer system has to be designed with safety in mind. Identifying and assessing hazards is not enough to make a system safe. For the Safety Critical Systems, minimizing the project risks, improving and guaranteeing the product quality, project cost and improved communication among stakeholders and other involved parties like the developing team, the acquirer and the supplier as well are considered as the main factors to achieve the intended project in a perfectly managed way. Thus, The V Model-XT is suitable to design safety critical systems because this model and its extended tailoring property provides all the following mentioned objectives

- **Minimize the Project Risks**

The V – Model XT structures improve transparency as well as the ability to plan projects because it divides projects into many types and therefore each type will have a different way of approach and developing.

- **Improve and guarantee for the Product Quality**

The V Model XT inherits from its ancestor the ability to validate in the early stages. For any error that counters, the error should be fixed early to secure the quality. Thus, it is inevitable to say that the quality of the product would be ensured.

- **Reduce the Entire Project Cost and System Life Cycle Cost**

Thanks to the well – defined structures, all the stages from development, operation, testing to maintenance can be calculated and controlled by the model. Moreover, with the ability to track down error in early stages, it can reduce a significant cost and also time.

- **Improve the communication between different Stakeholders**

Since the validation could be made during the verification stages, especially for the Safety Critical Systems, there will be a huge requirement of communication between the testing team and the developing team. Also, the acquirer and supplier have to communicate to agree on the system requirement.

Thus, we can say that the V – Model XT is a good model, especially for Safety Critical System, for guidance in processing, creating contracts and can be used as the base communication.

d) Agile Process Models

State the advantages and disadvantages of Agile Process Models for SCS developments!

Agile methods were introduced as an alternative to traditional methodologies, which had a lot of documentation and were too restrictive when dealing with changing requirements. In response to these concerns agile methods offer a more relaxed approach towards documentation and provide a flexible development lifecycle based on short iterations.[7]

- **Advantages of Agile Process Model for SCS developments**

Douglass [3] identified a list of advantages of Agile Process Model for SCS developments:

- Improve quality
- Customer acceptability.
- Decreased development costs
- Reduced documentation
- **Disadvantages of Agile Process Model for SCS developments**

According to Tordrup, Lise & Nielsen, Peter[8] The analysis of the challenges of agile software development in a safety-critical context showed that the literature focuses on four problematic practice areas understood as issues to deal with the following:

- Light documentation.
- Flexible requirements written in user stories.
- Iterative and incremental life cycles.
- Test-first process.

Are there possibilities in the V-Model XT to include Agile Processes like SCRUM and XP? How can this be done in practice?

Yes. It is possible to include Agile processes like SCRUM and XP in the V-Model XT. Below we present an approach to integrate a concrete agile method, the Microsoft Solution Framework 4.0 for Agile Software Development, into the V-Model XT.

The Microsoft Solution Framework (MSF) 4.0 is a process framework similar to the V-Modell XT addressing software development. The MSF framework contains the metamodel and can be instantiated into one or more prescriptive processes.[9]

The MSF contains a set of fundamental principles: a team model, a process model and disciplines. The basic process model is an iterative approach. The structure and the contents of the MSF are based on a set of principles of a particular project. Some – but not all – principles are: “Stay agile and expect change” or “Always create ship-pable products” [9].

The MSF only contains three basic elements: Roles, Activities and Work Streams. Roles are defined in a non-hierarchical team model - [9] speaks of peers. Work streams are assigned to roles. A work stream is a sequence of activities, such as working on the project plan. Each activity has products and roles assigned. The MSF Agile includes a simple, spiral-like iterative model for project operation.[9]

There are 2 ways to connect MSF to the V Model XT.

1st Way - Extension: The V-Model contains the capabilities for integrating additional functionality. By adding additional process modules, new products, roles and activities may be introduced. New operation strategies can be added as well. An additional process module addressing the MSF contents might contain the role release manager, the product prototype and the topic scenario. This way should be the preferred one if compatibility with V-Modell XT is required.[10]

2nd Way - Specialization: The second way is to alter the V-Modell XT itself. This means creating an organization-specific V-Modell XT derivative. An additional process module has to be created, too. But MSF contents, comparable to existing V-Modell ones, can be merged directly. As a consequence, standard V-Modell contents may be replaced by MSF contents, but some generic contents might be lost. [10]

Reference

1. SAFETY CRITICAL SYSTEMS IN MEDICINE M.SPANDAN & J.VARUN CHANDRA
December 6, 2011, https://www.academia.edu/4173741/Safety_Critical_Systems_in_medical_Field
 - 2.D-Turgay-Altılar-2/publication/283197222_Test-Driven_Approach_for_Safety-Critical_Software_Development/links/5706580008aea3d280210fce/Test-Driven-Approach-for-Safety-Critical-Software-Development.
 - 3.Tucker, Allen and Belford, Geneva G.. "Computer science". *Encyclopedia Britannica*, 1 Sep. 2020, <https://www.britannica.com/science/computer-science>. Accessed 4 May 2021.
 - 4.<http://poincare.math.rs/~vlada/Courses/Matf%20MNSR/Literatura/Scientific%20Methods%20in%20Computer%20Science.pdf> Scientific Methods in Computer Science Gordana Dodig-Crnkovic
Department of Computer Science Mälardalen University Västerås, Sweden.
- [***]. Lewis R. Tucker et al., "A Multidimensional Assessment of Ethical Codes: The Professional Business Association Perspective", *Journal of Business Ethics*, vol. 19, pp. 287-300, 1999
5. Rausch, Andreas & Bartelt, Christian & Ternité, Thomas & Kuhrmann, Marco. (2005). The V-Modell XT Applied-Model-Driven and Document-Centric Development.
 6. Projekt WEIT - Weiterentwicklung des Entwicklungsstandards für IT-Systeme des Bundes auf Basis des V-Modell-97, <http://www.v-modell-xt.de>, 2003
 7. Douglass, B.P., and Ekas, L.. Adopting agile methods for safety-critical systems development. IBM, 2012.
 8. Tordrup, Lise & Nielsen, Peter. (2018). A Conceptual Model of Agile Software Development in a Safety-Critical Context: A Systematic Literature Review. *Information and Software Technology*. 10.1016/j.infsof.2018.06.004.
 - 9.. Kuhrmann, Marco & Ternité, Thomas. (2006). Including the Microsoft Solution Framework as an agile method into the V-Modell XT.
 10. Richard Hundhausen, Working with Microsoft Visual Studio 2005 Team System. Microsoft Press, ISBN 0-7356-2185-3, 2005.

Assignment 4

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

1. Ainan Ahmed Chowdhury--1377020
2. Shrabanti Saha Rimi--1377509
3. Ashis Banik -- 1377253
4. Aktari Sadia Sabah--1382413
5. Syed Fawzul Azim--1364224

Question-1

a) Summarize the main contents of SWEBOK V3.0

Answer:

Main contents of SWEBOK V3.0

SWEBOK stands for Software Engineering Body of Knowledge, and it is an international standard guideline for software engineers. SWEBOK V3.0 is the most recent fully revised and modified edition of the widely recognized Guide to the Software Engineering Body of Knowledge. This work is in partial fulfillment of the Society's responsibility to promote the advancement of both theory and practice for the profession of software engineering. SWEBOK V3.0 is now deliberately developed to be regularly reviewed and updated as technologies and the engineering profession evolve over time, ensuring that it remains continuously applicable. this Guide does not present the entire the body of knowledge for software engineering but rather serves as a guide to the body of knowledge that has been developed over more than four decades

The objectives of the Guide to the Software Engineering Body of Knowledge project are to

- characterize the contents of the Software Engineering Body of Knowledge;
- provide a topical access to the Software Engineering Body of Knowledge;
- promote a consistent view of software engineering worldwide;
- clarify the place of, and set the boundary of, software engineering with respect to other disciplines such as computer science, project management, computer engineering and mathematics;
- provide a foundation for curriculum development and individual certification and licensing material.

SWEBOK V3.0 holds 15 Knowledge Areas, plus a new Appendix on Standards in Software engineering.

15 SWEBOK Knowledge Areas
1. Software Requirements
2. Software Design
3. Software Construction
4. Software Testing
5. Software Maintenance
6. Software Configuration Management
7. Software Engineering Management
8. Software Engineering Process

9. Software Engineering Models and Methods
10. Software Quality
11. Software Engineering Professional Practice
12. Software Engineering Economics
13. Computing Foundations
14. Mathematical Foundations
15. Engineering Foundations

b) Requirement analysis is concerned with the process of analyzing requirements to

- Detects and resolves conflicts between requirements.
- Discover the bounds of the software and how it must interact with its organizational and operational environment.
- Elaborate system requirements to derive software requirements.

Requirements Classification:

Requirements can be classified on a number of dimensions. Examples include the following:

- Whether the requirement is functional or nonfunctional.
- Whether the requirement is derived from one or more high-level requirements or an emergent property.
- Whether the requirement is on the product or the process.
- The requirement priority.
- The scope of the requirement. Scope refers to the extent to which a requirement affects the software and software components.
- Volatility/stability. Some requirements will change during the life cycle of the software—and even during the development process itself.

Conceptual Modeling:

The development of models of a real-world problem is key to software requirements analysis. Their purpose is to aid in understanding the situation in which the problem occurs, as well as depicting a solution. Hence, conceptual models comprise models of entities from the problem domain, configured to reflect their real-world relationships and dependencies.

The factors that influence the choice of modeling notation include these:

- The nature of the problem. Some types of software demand that certain aspects be analyzed particularly rigorously
- The expertise of the software engineer.
- The process requirements of the customer

Requirements Negotiation:

This concerns resolving problems with requirements where conflicts occur between two stakeholders requiring mutually incompatible features, between requirements and resources, or between functional and nonfunctional requirements.

Formal Analysis:

Formal analysis has made an impact on some application domains, particularly those of high integrity systems. The formal expression of requirements requires a language with formally defined semantics. The use of a formal analysis for requirements expression has two benefits. First, it enables requirements expressed in the language to be specified precisely and unambiguously, thus (in principle) avoiding the potential for misinterpretation. Secondly, requirements can be reasoned over, permitting desired properties of the specified software to be proven.

Question-2

a) Browse for alternative literature!

We already browse for alternative literature of Generic Safety Standard IEC61508 and we get an Extended Systematic Literature. We summarize it and discuss the process model in detail.

b) Summarize the main aspects of the standard!

The overall title of IEC 61508 is; “Functional safety of electrical, electronic and programmable electronic (E/E/PE) safety-related systems”[1].

The IEC 61508 consists of the following 7 parts:

1. General requirements.
2. Requirements For RECIPES.
3. Software requirements.
4. Definitions.
5. Methods for the determination of SIL.

6. Guidelines on the application of part 2 and 3.
7. Overview of techniques and measures.[2]

IEC 61508 is both a stand-alone standard and can also be used as the basis for sector and product standards. In its latter role, it has been used to develop standards for both the process and machinery sectors and is currently being used to develop a standard for power drive systems. It has influenced, and will continue to influence, the development of E/E/PE safety-related systems and products across all sectors.[1]

The requirements of the ICE 61508 could be summarized as follows:

Range and extent of measures and techniques for the avoidance and control of faults (HW and SW) applied during the design and development including functional safety management during the whole life cycle,

Hardware fault tolerance of subsystems (structure) in combination with safe failure fraction and diagnostic coverage (internal self-tests),

Probability of failure to danger of the subsystems by reliability modeling techniques,

Measures and techniques for avoidance and control of faults during the design and development of the application software.[2]

IEC 61508 is mainly concerned with E/E/PE safety related systems whose failure could have an impact on the safety of persons and/or the environment. However, it was recognized that the consequences of failure could have serious economic implications and in such cases the standard could be used to specify any E/E/PE system used for the protection of equipment or product;[1]

IEC 61508 uses three safety lifecycles in order that all relevant phases are addressed. They are:

- The Overall Safety Lifecycle.
- The E/E/PES Safety Lifecycle.
- The Software Safety Lifecycle.

c)Which process model is proposed?

Agile development strategies, for example, Scrum are utilized in an ever-increasing number of spaces of software development. For safety-critical systems, however, safety engineers feel that agile development does not fit, the main reason being that these projects require a strict plan which makes later changes costly. Agile development is inconsistent with acceptable practices for developing safety-critical software. The most ordinarily referenced issues are plan – e.g., structural plan and low-level plan – and arranging in addition to rules for refactoring and traceability. On the positive side, the most normally referenced positive thoughts are constantly testing and flexible.[3]

The two principal challenges are the standards' requirements on detailed planning and requirements for proof of conformance. The Safe Scrum development cycle will be a significant contribution to this work. This will lead to a process that is better adapted to handle changes that occur in any software development process and give us an incremental process – development, testing, and verification – that again will lead to more efficient software development.[4]

Elements of the agile development process are already used or considered for use in several important industrial domains such as automotive and air traffic management (used) and avionics and industrial automation (planned or evaluated for use). The solution suggested is Safe Scrum is better than other processes, since they have kept more of the agile process concepts. This is important in order to reap the maximum benefits from using Scrum.

Reference

1. Bell, Ron. (2006). Introduction to IEC 61508. 3-12.
2. H. Gall, "Functional safety IEC 61508 / IEC 61511 the impact to certification and the user," 2008 IEEE/ACS International Conference on Computer Systems and Applications, 2008, pp. 1027-1031, doi: 10.1109/AICCSA.2008.4493673.
3. The application of Safe Scrum to IEC 61508 certifiable software Tor Stålhane^a*, Thor Myklebust^b, Geir Hanssen^b ^a NTNU, Trondheim, Norway ^b SINTEF ICT, Trondheim, Norway.
4. Introduction to IEC 61508 Ron Bell Health & Safety Executive Bootle, UK.

Assignment 5

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

1. Ainan Ahmed Chowdhury--1377020
2. Shrabanti Saha Rimi--1377509
3. Ashis Banik -- 1377253
4. Aktari Sadia Sabah--1382413
5. Syed Fawzul Azim--1364224

1. Lab Project

Research and explain the Apple IOS/Google Android framework underlying the Corona Warn App

As a response to the COVID-19 pandemic, digital contact tracing has been proposed as a tool to support the health authorities in their quest to determine who has been in close and sustained contact with a person infected by the coronavirus. In April 2020 Google and Apple released the Google Apple Exposure Notification (GAEN) framework, as a decentralised and more privacy friendly platform for contact tracing. The GAEN framework implements exposure notification mostly at the operating system layer, instead of fully at the application layer[1]. Although GAEN is a joint framework, there are minor differences in how it is implemented on Android (Google) and iOS (Apple). GAEN works on Android version 6.0 (API level 23) or higher, and on some devices as low as version 5.0 (API level 21) [2]. On Android GAEN is implemented as a Google Play service. GAEN works for Apple devices running iOS 13.5 or higher.

At the Bluetooth and 'cryptographic' layer GAEN works the same on both platforms however. This implies that ephemeral proximity identifiers sent by any Android device can be received and interpreted by any iOS device in the world and vice versa. In other words: users can in principle get notified of exposure to infected people independent of the particular operating system their smartphone runs, and independent of which country they are from. (In practice some coordination between the exposure notification apps and the back-end servers of the different health authorities involved is required.)[1]

As an optimisation step, devices do not randomly generate each and every ephemeral proximity identifier independently. Instead, the ephemeral proximity identifier Id_i to use for a particular interval i on day d is derived from a temporary exposure key K_d (which is randomly generated each day) using some public deterministic function f . In other words $Id_i = f(K_d, i)$. With this optimization, devices only need to store exposure keys in S , as the actual ephemeral proximity identifiers can always be reconstructed from these keys.

Generating, broadcasting, and collecting ephemeral proximity identifiers happens automatically at the operating system layer, but only if the user has explicitly enabled this by installing an exposure notification app and setting the necessary permissions,[3] or by enabling exposure notifications in the operating system settings.[4] Apple and Google do not allow exposure notification apps to access your device location.[5] By default, exposure notification is disabled on both platforms. When enabled, the database of S of exposure keys and the database R of identifiers received are stored at the operating system layer, which ensures that data is not directly accessible by any app installed by the user.

Actual notifications are the responsibility of the exposure notification app. In order to use the data collected at the operating system layer, the app needs to invoke the services of the operating system through the GAEN Application Programming Interface (API). Apps can only access this API after obtaining explicit permission from Google or Apple. The API offers the following main functions.

- Retrieve the set of exposure keys (stored in S). “The app must provide functionality that confirms that the user has been positively diagnosed with COVID-19.” But this is not enforced at the API layer. In other words, the app (once approved and given access to the API) has access to the exposure keys.
- Match a (potentially large) set of exposure keys against the set of ephemeral proximity identifiers received from other devices earlier (stored in R), and return a list of risk scores (either a list of daily summaries, or a list of individual < 30 minute exposure windows). This function is rate limited to a few calls per day.

The API also ensures that the user is asked for consent whenever an app enables exposure notification for the first time, and whenever user keys are retrieved for upload to the server of the health authorities after the user tested positive for COVID-19. The API furthermore offers functions to tune the computation of the risk scores.

Based on the last point present your current list of requirements

Process Overview:

We are going to work on Corona warn App where we want to add some extra features. Recently vaccination of Covid-19 has already started. We want to add vaccine information to this app so that people can easily complete the registration process for vaccination at home. We also try to provide previous and present data of Covid-19 cases for each federal state and each district of Germany. This process is fully dynamic, every data will be filtered by weeks.

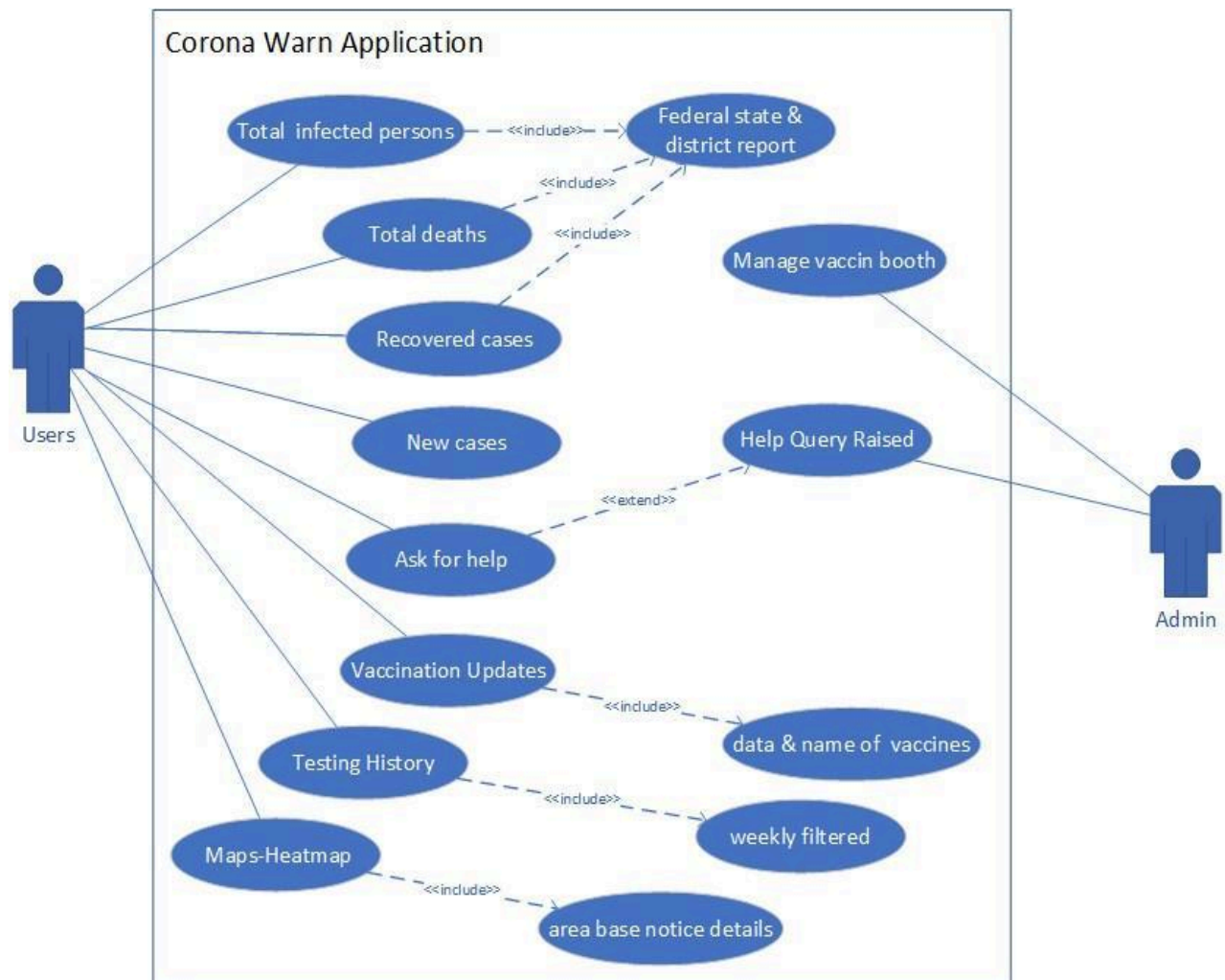
Process Actors:

- 1.Admin
- 2.User

Functional Requirement:

1. We want to show total cases of infected persons, the total number of deaths, totally recovered cases of Covid-19 for each federal state and each district of Germany for previous days and we also show a new number of infected persons, new deaths, new recoveries.
2. Through our project, we will provide the Vaccination History-number of vaccination on a particular date/in X-days, vaccination data (total dose administered, complete vaccinated/only one dose). Also, we will provide data for individual vaccines (i.e., j&#amp;j, Mjoderna, AstraZeneca,Biotech).
3. Testing History- Number of PCR-tests executed, positive PCR tests, rate of positive PCR which can be filtered by weeks.
4. We provide Maps-Heatmap (PNG) of individual state and district which will show the important incidents such as lockdown notice, movement pass based on area.[6]

Deliver the work-products of this phase! Especially: Use Cases for your application



Please review the most important estimation methods and tailor them for your specific project, especially – the Function Point method or a derivative, and – COCOMO , Present your results!

Components of Function Point Analysis

Function Point Analysis Part-1:

Determine the following components:

Types of FP Attributes

Measurements Parameters	Examples
1.Number of External Inputs (EI)	Input screen and tables
2. Number of External Output (EO)	Output screens and reports
3. Number of external inquiries (EQ)	Prompts and interrupts.

4. Number of internal files (ILF)	Databases and directories
5. Number of external interfaces (EIF)	Shared databases and shared routines.

Analysis of the software system as presented in the user point of view.

The number of various components:

- 1.External Inputs (EI): 0
- 2.External Outputs (EO): 3
- 3.Inquiries (EQ): 8
- 4.Internal Logic File (ILF): 4
- 5.External Logic File (ELF): 8

The degree of complexity (Simple, Average, Complex) was evaluated for each component.

Function Point Analysis Part-2:

Compute the unadjusted function point (UFP)

- 1.rate each component as low average or high.
- 2.for transactions (EI, EO, EQ), the rating is based on the FTR, and DET. [FTR- The number of files updated or referenced, DET- The number of user recognizable fields]

Number of user recognizable fields based on the table below an EI that references 0 files and 10 user recognizable fields would be ranked as Low.

File Type Referenced(FTR)	Data Element Type(DET)		
	1-4	5-15	15+
0-1	L	L	A
2	L	A	H
3+	A	H	H

For files (ILF and ELF), the rating is based on the RET and DET.

RET- The number of users-recognizable data elements in an ILF or ELF.

DET- The number of users-recognizable fields.

Based on the table below an ILF that contains 4 data elements and 6 user recognizable fields would be ranked as Low. [7]

Record Element Type(RET)	Data Element Type(DET)		
	1-19	20-51	51+
1	L	L	A
2-5	L	A	H
6+	A	H	H

Convert ratings into UFP's (Unadjusted Function Points)

1. Number of external inputs (EI)	3	4	6
2. Number of external outputs (EO)	4	5	6
3. Number of external inquiries (EQ)	3	4	5
4. Number of internal files (ILF)	7	10	15
5. Number of external interfaces (EIF)	5	7	10
1. Number of external inputs (EI)	3	4	6

Program Characteristic	Function Points		
	Low Complexity	Medium Complexity	High Complexity
External Inputs	$0 \times 3 = 0$	4	6
External Outputs	$3 \times 4 = 12$	5	7
External Queries	$3 \times 3 = 9$	4	6
Internal Logical Files	$7 \times 7 = 49$	10	15
External Interface Files	$5 \times 5 = 25$	7	10
Unadjusted Function Point total			95

Function Point Analysis Part-3:

Compute Value Adjustment Factor (VAF) based on 14 general system characteristics (GSC). Weight each GSC on a scale of 0 to 5 based on whether it has no influence to strong influence.

- Data communications-3

- Distributed data processing-0
- Performance-0
- Heavily used configuration-2
- Transaction rate-3
- On-Line data entry-0
- End-user efficiency-3
- On-Line update-3
- Complex processing-4
- Reusability-4
- Installation ease-1
- Operational ease-3
- Facilitate change-2
- Multiple sites-5

Total VAF-33

Compute the FP as follows:

VAF=Sum (GSC)

FP=UFP*(0.65+(VAF*0.01))

FP=95*(0.65+(33*0.01)) =93.1

FP=93

Convert FP to line of source code (SLOC)

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
Brio +	14	14	13	16
C *	97	99	39	333
C++ *	50	53	25	80
C# *	54	59	29	70

95UFP* 25(C++)SLOC/UFP=2375SLOC=2KLOC

COCOMO

Basic COCOMO Model: [8]

It estimates the software roughly and quickly. It is mostly for small – medium sized software.

ORGANIC:

Effort = a(KLOC)^b Person-month

= 2.4(2)^{1.05} Person-month

=5 Person-month

Development-Time= c(Effort)^d Months

=2.5(4)^{0.38} Months

=4 Months

Safety standards and modern process models: Please summarize the Master's Thesis of Fredriksen:

Explain in your own words, why the (R)UP is useful for the development of SCS

Like most methodologies, RUP lifecycle is broken into four main phases (cycles), each phase working on a new generation of the product:

1. Inception phase 2. Elaboration phase 3. Construction phase 4. Transition phase

1. Inception phase Essentially, in this cycle, your team determines the structure and the basic idea of the project. Also, the team will decide if the project is worth pursuing at all based on the estimated costs, the necessary resources and the goal they are trying to achieve with the project.

2. Elaboration phase The aim of this phase is to analyze the requirements and the architecture of the system, develop the project plan and eliminate the highest risk elements of the project. It's undoubtedly the most critical of all stages as it signifies the transition from low-risk to high-risk. It's also the point when your team has to make a decision whether to start a construction (development and coding) or not.

3. Construction phase At this stage, your team is finally ready to develop all components and features and integrate them into the product. It's a manufacturing process where your team focuses on managing resources in order to optimize costs, schedules and the quality.

4. Transition Phase The transition phase is the moment when the product is finally finished, released and delivered to customers. However, once the product is given to the user, there are a number of issues that can arise. This requires the team to handle all the bug-fixes and correct problems, or to finish some features that were postponed. At the end of each phase, there is an important Project Milestone - a point in time when your team confirms that certain goals have been achieved.

Explain the difficulties in merging the (R)UP and the life cycle requirements of IEC61508

1. It allows you to deal with changing requirements regardless of whether they are coming from the customer or from the project itself.
2. It emphasizes the need for accurate documentation.
3. It forces integration to happen throughout the software development, more specifically in the construction phase.

Difficulties in merging the RUP and the lifecycle requirements of IEC61508:

The RUP provide a process framework with the capability of customization in software engineering; frameworks for defining vast spectrum of different size, complexities and considerations projects. This concern provides the potential to produce software based on reduced risk and encounter main problems which leads to a reduction in cost and increase in potential success, hence an advantage. At this stage there does not exist any methodology to expand safe software from developing Safety-Critical systems based on objective orientation, axial architecture capable of gradual expansion and repetitiousness. Attempt is made in this article to apply RUP with respect to the safety rules printed in IEC 61508, in order to define and customize the necessities of Safety-Critical systems.

IEC61508 has a big number of requirements but all of them are not related to process, we need to distinguish those necessities in IEC61508 applicable for RUP. This is done by defining different selection criteria to limit the requirements to process requirements. Process requirements are requirements that constraints upon the development process of the system. We might divided The development requirement process into two classes: the requirements

How did Frederiksen combine the approaches of IEC61508 and (R)UP?

The Rational Unified Process (RUP) described is a software engineering process developed and marketed by Rational Software. It provides a disciplined approach to assigning and managing tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end users within a predictable schedule and budget.

The evaluation of IEC61508 and RUP seeks to identify the requirements of a development process that defines all relevant safety activities, explores synergy effects, applies to requirements from customer, government and certification authorities, defines roles and responsibility related to safety, defines safety specific milestones and is acceptable with regards to time and economic constraints - a process that also should be scalable.

Examples:

- Hazard and risk analysis.
- Determine the safety integrity level for safety instrumented functions.
- Interpretation problem of risk parameters

Reference

1. Jaap-Henk Hoepman. (2021). A Critique of the Google Apple Exposure Notification (GAEN) Framework.
2. L. van Dorp et al. "Emergence of genomic diversity and recurrent mutations in SARS-CoV-2". *Infection, Genetics and Evolution* 83 (2020), p. 104351.

3. S. Vaudenay. "Analysis of DP3T". Cryptology ePrint Archive 2020/399 (2020).
4. M. Veale. "Privacy is not the problem with the Apple-Google contacttracing toolkit". The Guardian (July 1, 2020).
5. M. Veale. "Sovereignty, privacy and contact tracing protocols". In: L. Taylor, G. Sharma, A. Martin, and S. Jameson. Data Justice and COVID-19: Global Perspectives. London: Meatspace Press, 2020, pp. 34–39.
6. Guide to the Software Engineering Body of Knowledge Version 3.0 SWEBOK® A Project of the IEEE Computer Society, Editors Pierre Bourque, École de technologie supérieure (ÉTS) Richard E. (Dick) Fairley, Software and Systems Engineering Associates (S2EA)
7. 2013 IEEE International Conference on Computational Intelligence and Computing Research, DOI: 10.1109/ICCIC.2013.6724240
8. S. Sabrjoo, M. Khalili and M. Nazari, "Comparison of the accuracy of effort estimation methods," 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2015, pp. 724-728, doi: 10.1109/KBEI.2015.7436134.

Assignment 6

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

1. Ainan Ahmed Chowdhury--1377020
2. Shrabanti Saha Rimi--1377509
3. Ashis Banik -- 1377253
4. Syed Fawzul Azim--1364224

1. Computer or Smart Phone Application for COVID-19 Assistance Simulation:

- **Prepare a first list of detailed requirements and write formal Use Case texts!**
- **Please give a detailed explanation of the general architecture of the problem space, using UML diagrams, e.g. Sequence, Collaboration and State Transition diagrams!**
- **Please prepare a detailed Project Plan!**
- **Plan and update frequently your project estimation, f.i. using Function Points and/or COCOMO II, and schedule!**

Project Title --> COVID-19 Assistance Application

Description

Nowadays Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people who fall sick with COVID-19 will experience mild to moderate symptoms and recover without special treatment. During the present circumstance, it's truly critical to spread the data like- - vital information of the COVID-19 patients' amount report, imperative guideline and rules for various urban areas, the all-out number of vaccinated patients' report, and the quantity of doses portions given in every area, and so forth This framework Can't assemble the all-essential data of the COVID-19 patients' and furthermore didn't spread the appropriate data of the clients. Assuming the framework has an appropriate API (Database Information), and add those highlights it will be an ideal one. In this way, we make a COVID-19 Assistance System utilizing API (Database Information) and add those highlights. At the point when any client needs to know frequently any assistance from this application then there will be ongoing data of COVID-19 patients' that the client will discover any of this without any problem. As technology is spreading throughout the world the automated system will also take the place of the manual system. This project provides the customer suggestion and information based on their search.

1.Product and Process Requirements:

Product Requirements:

1. Historical data for Germany, each federal state and each district. Historical Data- total cases, total deaths, total recovered, cases per age group (data can filtered to show last X-days data)
2. New infections, new deaths, new recoveries (difference from the previous day)
3. Vaccination Data- total dose administered, complete vaccinated/only one dose
 - 3.1. Vaccination data from individual vaccines (i.e. j&j, moderna, astraZeneca, Biontech)
 - 3.1.2. Vaccination History- number of vaccination on a particular date/in X-days
4. Testing History- Number of PCR-tests executed, positive PCR tests, rate of positive PCR (can also be filtered by weeks)
5. Maps - Heatmap (PNG) of week incidences for districts.

5.1. Shows the vital rules and regulations of week incidences for districts.important incidents such as lockdown notice, movement pass based on area

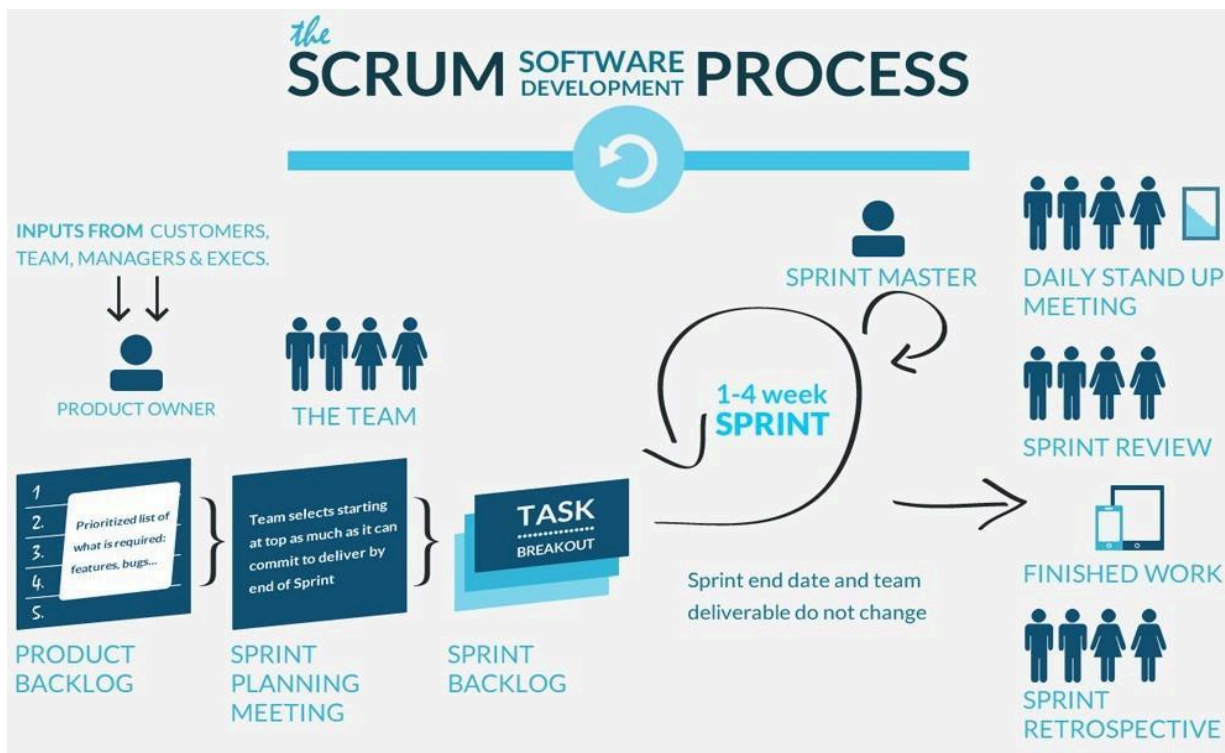
Process Requirements:

The objective of this process model is to provide an understanding that the requirements process is not a discrete front-end activity of the software life cycle, but rather a process initiated at the beginning of a project that continues to be refined throughout the life cycle.

1. Identifies software requirements as configuration items and manages them using the same software configuration management practices as other products of the software life cycle processes;

2. Needs to be adapted to the organization and project context.

We will be using agile scrum in this project; this methodology is a project management system that relies on incremental development. Each iteration consists of two- to four-week sprints, where each sprint's goal is to build the most important features first and come out with a potentially deliverable product. More features are built into the product in subsequent sprints and are adjusted based on stakeholder and customer feedback between sprints.



2.Functional and Nonfunctional Requirements:

Business Requirement Specification [BRS]

Action 1: BR's are entered in the system

Action 2: BR's are reviewed by the Business analysts (Trainees) to give them a meaning full understanding.

SL	BRS Title	Description
BR 1	Maintain system with organization structure	Set up system Level features and control with access rights and groups for users to manage the different modules.
BR 2	Maintain Historical Data system	Setup Historical Data system for COVID-19 patients' report.
BR 3	Maintain Authentic New (infections, deaths, recoveries) information.	Setup configuration for new information.
BR 4	Maintain Vaccination Data	Setup configuration for amount of Vaccination
BR 5	Maintain Number of PCR	Setup configuration for PCR-tests result (can also be filtered by weeks).
BR 6	Maintain Maps incidences for districts.	Record Heatmap (PNG) of week with the system as per incidences for districts.
BR 7	Maintain Reports & Dashboards	Configure Reports and dashboard as per users need

3.Requirement Elicitation

Stakeholder / User Identification:

This project is to develop as a product with reference to competitive system, the major Actors / User of this System Shall be

- 1.Admin User
- 2.End User
- 3.3rd party user

Task of Admin User:

- 1.Log in into the system.
- 2.Admin have all power of the system

Task of End User:

1. View Historical Data- total cases, total deaths, total recovered, cases per age group.
2. View New infections, new deaths, new recoveries.
3. View Vaccination Data- total dose administered, complete vaccinated.
4. View Testing History Number of PCR-tests executed, positive PCR tests, rate of positive PCR (can also be filtered by weeks).
5. Maps - Heatmap (PNG) of week incidences for districts.

Task of 3rd party user:

1. View API

4. User Need Elicitation:

Action 1: The user, once identified, has passed through the elicitation techniques (Interview / Brainstorm / QFD / Use case / FAST) to provide the user needs.

Elicitation Technique:

For this project we chose to do elicitation by interview.

So, why opt for an interview above other techniques?

Interviews offer the analyst an opportunity to establish rapport and trust with the interviewee.

By conducting a face-to-face meeting, the analyst can start a cordial relationship with the interviewee to make them feel involved in the project.

Interviews allow the interviewee to respond freely and openly to questions, especially when the location is private.

Interviews provide an opportunity for the analyst to ask follow-up questions or re-word the question to get instant feedback from the interviewee.

Interviews present an opportunity for the analyst to observe non-verbal clues. It is not everything that an interviewee can put into words.

EXAMPLE:

- 1) How many people will be browsing at a time?
- 2) Do you want to update the software in future?

5. User Requirement Specification [URS]:

Action 1: The identified User needs are then reviewed by the business analyst to qualify them as a User Requirement under the category Normal / Expected / Exciting.

Action 2: The identified User Requirements are reviewed by the quality Expert to ensure their measurability (testable or not)

Action 3: The qualified User Requirements are reviewed by the customer / stakeholder / user and on qualification are treated as USER Requirement specification.

6. System Requirement Elicitation:

The user requirements are elicited using the elicitation techniques to identify The Functional Requirement, Non-Functional Requirement and System Requirement.

SL	Title	Description	Ref UR
FR 1	Add and List User		
FR 2	Add User Group, Define Access rights, assign users		
FR 3	Add and List Zip, Country, state, City		
FR 4	Add Branch of Historical Data		
FR 5	Add Branches of last X-days data		
FR 6	Edit system details		
FR 7	Add Groups of information		
FR 8	Add Vaccination Data		
FR 9	Add vaccination on a particular date, Assign rules		
FR 10	Add Number of PCR-tests, Assign Rules		
FR 11	Add positive PCR tests		
FR 12	Add Heatmap (PNG)		
FR 13	Setup Currency compatibility		
FR 14	Setup language compatibility		
FR 15	Setup Numeric Symbol		
FR 16	Setup Module to use		
FR 17	Setup Configuration for rules and regulations		
FR 18	Add and List User		
FR 19	Add User Group, Define Access rights, assign users		
FR 20	Admin Home page		
FR 21	Settings for master and ledger info		
FR 22	Settings for historical graphical screens		

SL	Title
NFR1	Add loading time is not more than 5 second
NFR2	Add performance time is 30 second fixed.
NFR3	Do not except any-critical path with functional requirement

Estimation Process & Tables:

Components of Function Point Analysis

Function Point Analysis Part-1:

Determine the following components:

Types of FP Attributes:

Measurements Parameters	Examples
1.Number of External Inputs (EI)	Input screen and tables
2. Number of External Output (EO)	Output screens and reports
3. Number of external inquiries (EQ)	Prompts and interrupts.
4. Number of internal files (ILF)	Databases and directories
5. Number of external interfaces (EIF)	Shared databases and shared routines.

Analysis of the software system as presented in the user point of view.

The number of various components:

- 1.External Inputs (EI): 0
- 2.External Outputs (EO): 3
- 3.Inquiries (EQ): 8
- 4.Internal Logic File (ILF): 4
- 5.External Logic File (ELF): 8

The degree of complexity (Simple, Average, Complex) was evaluated for each component.

Function Point Analysis Part-2:

Compute the unadjusted function point (UFP)

- 1.rate each component as low average or high.
- 2.for transactions (EI, EO, EQ), the rating is based on the FTR, and DET. [FTR- The number of files updated or referenced, DET- The number of user recognizable fields]

Number of user recognizable fields based on the table below an EI that references 0 files and 10 user recognizable fields would be ranked as Low.

File Type Referenced(FTR)	Data Element Type(DET)		
	1-4	5-15	15+
0-1	L	L	A
2	L	A	H
3+	A	H	H

For files (ILF and ELF), the rating is based on the RET and DET.

RET- The number of users-recognizable data elements in an ILF or ELF.

DET- The number of users-recognizable fields.

Based on the table below an ILF that contains 4 data elements and 6 user recognizable fields would be ranked as Low.

Record Element Type(RET)	Data Element Type(DET)		
	1-19	20-51	51+
1	L	L	A
2-5	L	A	H
6+	A	H	H

Convert ratings into UFP's (Unadjusted Function Points)

1. Number of external inputs (EI)	3	4	6
2. Number of external outputs (EO)	4	5	6
3. Number of external inquiries (EQ)	3	4	5

4. Number of internal files (ILF)	7	10	15
5. Number of external interfaces (EIF)	5	7	10
6. Number of external inputs (EI)	3	4	6

	Function Points		
Program Characteristic	Low Complexity	Medium Complexity	High Complexity
External Inputs	$0 \times 3 = 0$	4	6
External Outputs	$3 \times 4 = 12$	5	7
External Queries	$3 \times 3 = 9$	4	6
Internal Logical Files	$7 \times 7 = 49$	10	15
External Interface Files	$5 \times 5 = 25$	7	10
Unadjusted Function Point total			95

Function Point Analysis Part-3:

Compute Value Adjustment Factor (VAF) based on 14 general system characteristics (GSC).

Weight each GSC on a scale of 0 to 5 based on whether it has no influence to strong influence.

- Data communications-3
- Distributed data processing-0
- Performance-0
- Heavily used configuration-2
- Transaction rate-3
- On-Line data entry-0
- End-user efficiency-3
- On-Line update-3

- Complex processing-4
- Reusability-4
- Installation ease-1
- Operational ease-3
- Facilitate change-2
- Multiple sites-5

Total VAF-33

Compute the FP as follows:

$VPF = \sum (GSC)$

$FP = UFP * (0.65 + (VPF * 0.01))$

$FP = 95 * (0.65 + (33 * 0.01)) = 93.1$

FP=93

Convert FP to line of source code (SLOC)

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
Brio +	14	14	13	16
C *	97	99	39	333
C++ *	50	53	25	80
C# *	54	59	29	70

$95UFP * 25(C++)SLOC/UFP = 2375SLOC = 2KLOC$

COCOMO

Basic COCOMO Model:

It estimates the software roughly and quickly. It is mostly for small – medium sized software.

ORGANIC:

$Effort = a(KLOC)^b$ Person-month

$= 2.4(2)^{1.05}$ Person-month

$= 5$ Person-month

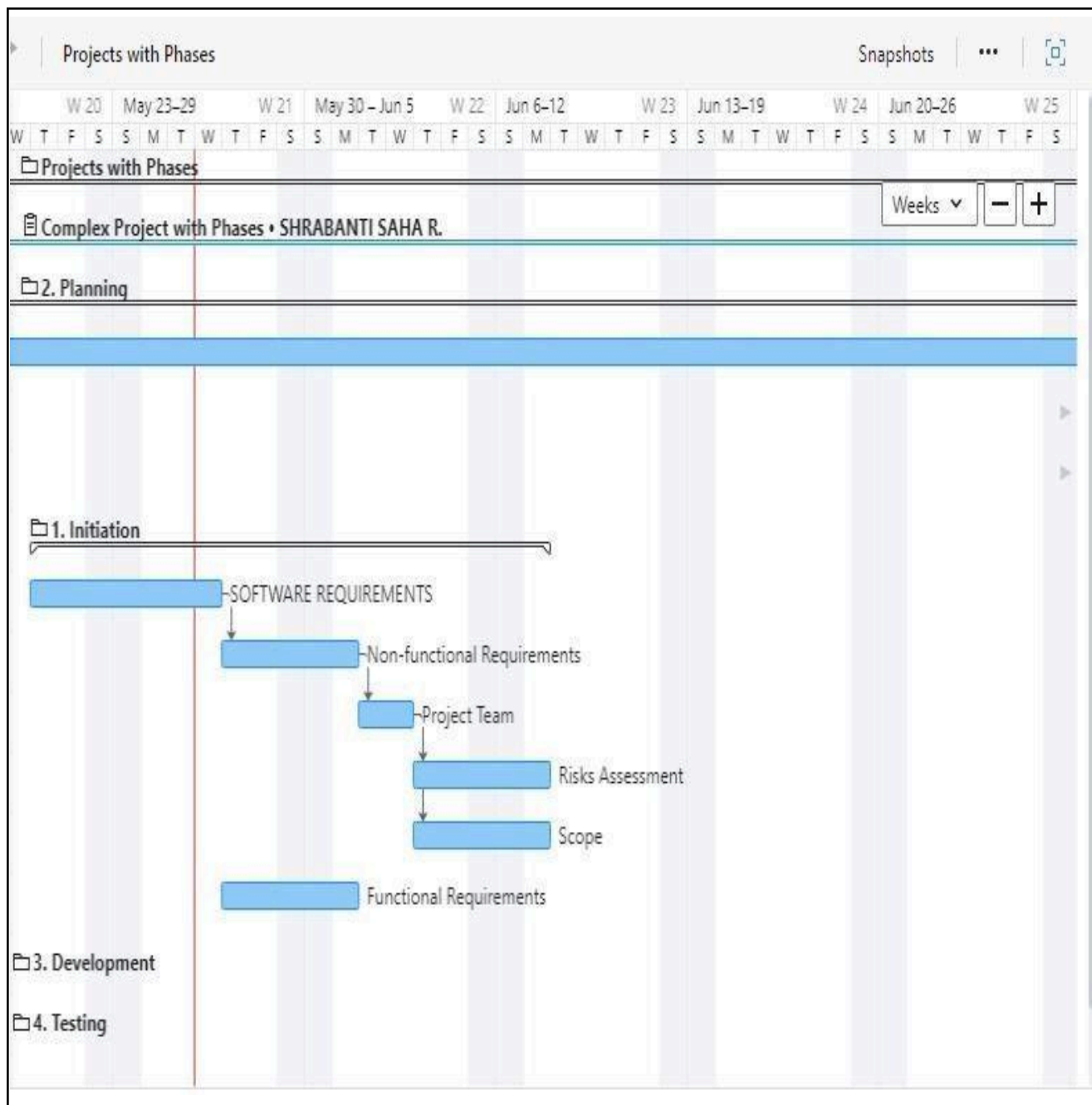
$Development-Time = c(Effort)^d$ Months

$= 2.5(4)^{0.38}$ Months

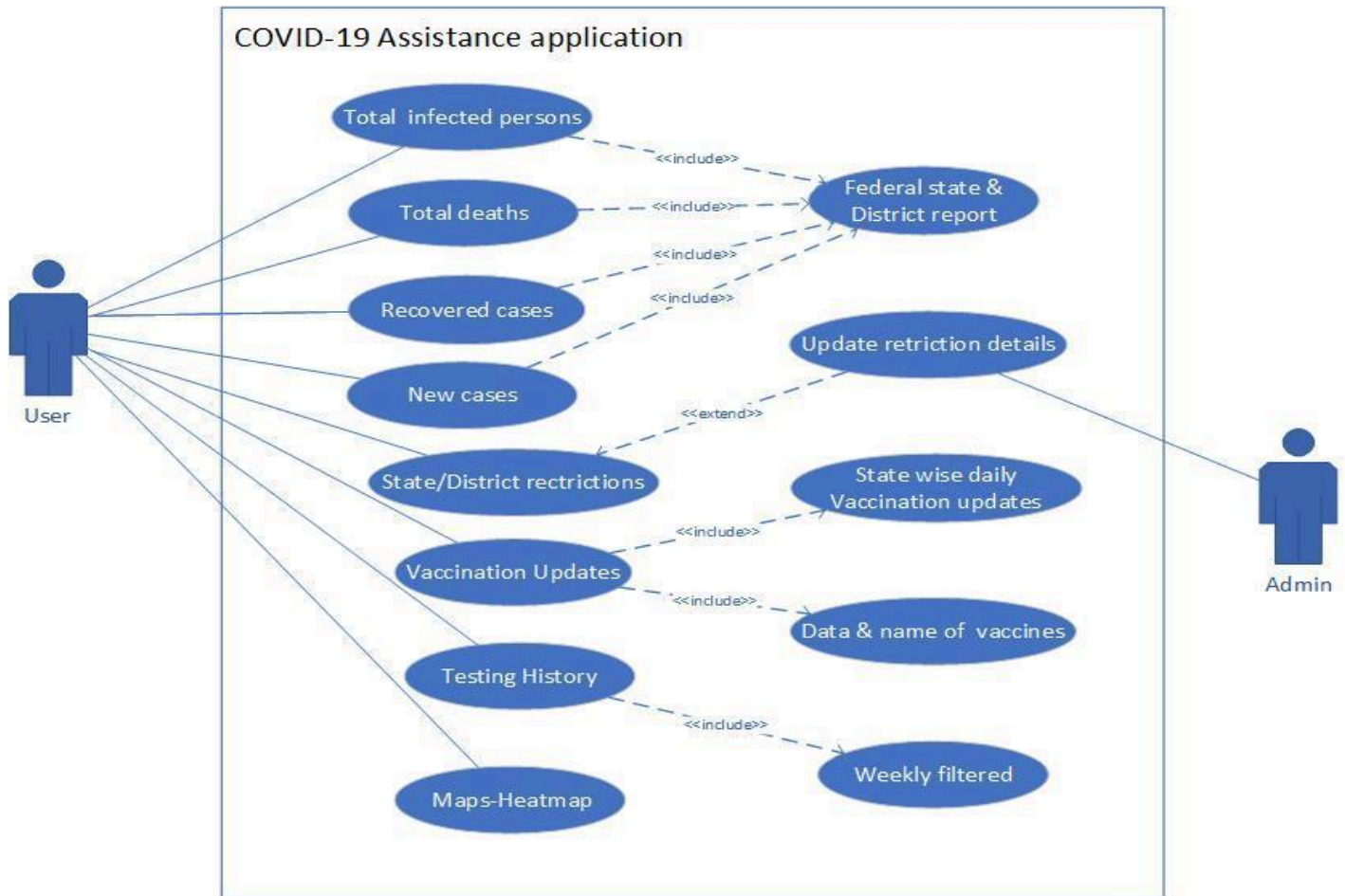
$= 4$ Months

Project Plan:

	Title	Start date	Due date	Predecessors ↓
1	▼ Projects with Phases			
2	▼ Complex Project wit...			
3	▼ 2. Planning			
4	Finalize proje...	05/05/2021	07/05/2021	
5	Identify requi...	07/06/2021	07/07/2021	4FS
6	Project plann...		07/05/2021	5FF
7	▼ 1. Initiation			
8	SOFTWARE R...	05/20/2021	05/26/2021	
9	Non-functio...	05/27/2021	05/31/2021	8FS
10	Project Team	06/01/2021	06/02/2021	9FS
11	Risks Assess...	06/03/2021	06/07/2021	10FS
12	Scope	06/03/2021	06/07/2021	10FS
13	Functional R...	05/27/2021	05/31/2021	
14	3. Development			
15	4. Testing			



Formal Use Case



Use case diagram

Use case	Check total infections
Actors	User
Actor Intentions	Users select the 'Total Infected persons' option & see the number of infections.
Cross References	Check data from API/Database.
System Responsibility	System shows the total infected person.
Alternative Courses	If User wants to see the infection number state/district wise they can see from the 'Federal state & District report' option.

Use case	Check Total deaths, New cases, Recovered cases
Actors	User
Actor Intentions	Users can select Total deaths/New cases/Recovered cases options to see the number of Total deaths/New cases/Recovered cases.
Cross References	Check data from API/Database.
System Responsibility	1.System checks all information & response to the required field. 2.Provide information to all users synchronously.
Alternative Courses	Users can see all sections state/district wise information.

Use case	Check State/District Restrictions
Actors	User
Actor Intentions	Users can see State/District wise restrictions. e.g. which city/state call or remove lockdown, new rules & restrictions, about social distance & mask rules, party rules, curfew rules etc.
Cross References	Check from Database.
System Responsibility	Check the updated news & provide to the uses.
Alternative Courses	

Use case	Vaccination updates & Testing history
Actors	User
Actor Intentions	Users open this field to gather knowledge about vaccination updates & testing history.
Cross References	API/Database.
System Responsibility	System verifies the Vaccination & testing data from the source and serves it.
Alternative Courses	1.User can check the date & name of vaccinations. 2.user can see weekly testing history. 3.User can check 'State wise daily Vaccination updates'.

Use case	Maps-Heatmap
Actors	User
Actor Intentions	Users can point out the intensity from the colored overlay on top of the map.
Cross References	API/Database.
System Responsibility	System check data from data server & show the current high & low spread areas on map.
Alternative Courses	

Use case	Update Restrictions details
Actors	Admin
Actor Intentions	Admin update all rules & regulations, restrictions, about social distance & mask rules, party rules, curfew rules etc.
Cross References	Database
System Responsibility	System updated information on database.

3. Real-Time Aspects:

- **Please elaborate on the Real-Time aspects of an COVID-19 Assistance System !**
- **Compare Hard- and Soft- Real-Time systems! Literature: Giorgio Buttazzo et al.: Soft Real-Time Systems. Springer 2005**

Real-Time Aspects:

Please elaborate on the Real-Time aspects of an COVID-19 Assistance System !

Real time aspect is very important for an COVID-19 Assistance system. For example here we will talk about the Exposure Notifications System created by Google and Apple. This system facilitate digital contact tracing during the COVID-19 pandemic. When used by health authorities, it augments more traditional contact tracing techniques by automatically logging encounters with other notification system users using their Android or iOS smartphone. For contract tracing, if data is not updated real time there is a high possibility of Spreading the virus.

Compare Hard- and Soft- Real-Time systems

A real-time system is a system that reacts to an event within a limited amount of time. So, for example, in a web page reporting the state of a Formula 1 race, we say that the race state is reported in real-time if the car positions are updated "as soon as" there is a change. In this particular case, the expression "as soon as" does not have a precise meaning and typically refers to intervals of a few seconds.[1]

Hard real time is a system whose operation is incorrect whose result is not produced according to time constraints. For example Air Traffic Control, Medical System, etc.

On the other hand, a Soft real time system is a system whose operation is degraded if results are not produced according to the specified timing requirement. For example Multimedia Transmission and Reception, Computer Games, etc.

The response time requirements of hard real-time systems are in the order of milliseconds or less and can result in a catastrophe if not met. In contrast, the response time requirements of soft real-time systems are higher and not very stringent. In a hard real-time system, the peak-load performance must be predictable and should not violate the predefined deadlines. In a soft real-time system, a degraded operation in a rarely occurring peak load can be tolerated. A hard real-time system must remain synchronous with the state of the environment in all cases. On the other hand soft real-time systems will slow down their response time if the load is very high. Hard real-time systems are often safety critical. Hard real-time systems have small data files and real-time databases. Temporal accuracy is often the concern here. Soft real-time systems for example, on-line reservation systems have larger databases and require long-term integrity of real-time systems. If an error occurs in a soft real-time system, the computation is rolled back to a previously established checkpoint to initiate a recovery action. In hard real-time systems, roll-back/recovery is of limited use.

4. Classical Hazard Analysis Methods

- **Please provide an overview of classical hazard analysis methods!**
- **Please look in detail at the Fault Tree Analysis method and give an explanation! The NASA handbook can be found on the server.²**
- **Please look at the structure of an FMEA!**
- **Please explore Event Tree Analysis too!**
- **Based on your Use Cases and your specification explore the use of a classical hazard analysis method!**

Classical Hazard Analysis Methods:

Hazard analysis is an examination of a system or subsystems to identify and classify each potential hazard that could occur in the system, and it must be carried out at an early stage of the system development. The aim of the analysis is to deliver a system which does not pose an unacceptable danger to its end-user or to the environment in which the system is installed [1] [2]. Here the hazard analysis involves three steps: (1) Deriving hazards from safety properties, (2) Using Fault Tree Analysis (FTA) to analyze the possible causes of each hazard, and (3) Converting each minimal cut-set of FTA into a formal property in terms of variables used in the formal specification. [3]

There are several hazard analysis methods some are as follows-

- Checklists
- Failure modes and effects analysis (FMEA)
- Failure modes, effects and criticality analysis (FMECA)
- Fault tree analysis (FTA)
- Management oversight and risk tree analysis (MORT)
- Event tree analysis (ETA)
- Cause-consequence analysis (CCA)
- Hazards and operability analysis (HAZOP)
- Fault hazard analysis (FHA)
- State machine hazard analysis (SMHA)

The Fault Tree Approach:

FTA can be simply described as an analytical technique, whereby an undesired state of the system is specified (usually a state that is critical from a safety or reliability standpoint), and the system is then analyzed in the context of its environment and operation to find all realistic ways in which the undesired event (top event) can occur. The fault tree itself is a graphic model of the various parallel and sequential combinations of faults that will result in the occurrence of the predefined undesired event.

It is also important to point out that a fault tree is not in itself a quantitative model. It is a qualitative model that can be evaluated quantitatively and often is. This qualitative aspect, of course, is true of virtually all varieties of system models. The fact that a fault tree is a particularly convenient model to quantify does not change the qualitative nature of the model itself.

After confirming the hazards for safety properties, these hazards need to be analysed in order to identify its potential causes. In this research, the analysis is conducted using FTA as: (1) it is a useful tool for reliability and safety analysis; (2) it is a top-down approach starting with an undesirable event, called a top event, and all the possible ways that the top event can happen could be determined; and (3) if there is a critical failure mode, all the possible ways that the mode can occur will be discovered [4].

The top event is a potential hazard that has been derived from safety properties. For each hazard, a fault tree needs to be developed and an undesired event needs to be defined. Then, the event is resolved into its immediate causes; this resolution of events continues until the basic causes are defined. The purpose of contributing to the fault tree is to obtain the minimal cut-sets. Each minimal cut-set is the smallest combination of component failures that would cause the top event to occur if the component failures all occur [5].

The Event Tree Analysis:

- Forward search to identify outcomes of events.
- It is necessary to know where to start.
- Event tree from left to right.
- Branches to two alternatives:
 - Upper branch successful performance
 - Lower branch failure
- Includes probabilities.
- Initial event characterized by frequency.
- Secondary events are probabilities.
- Elimination of impossible branches.

Based on our use case and specification, we have two actors: admin and user of the application. Here the user gets information about total covid infected records, the number of total deaths because of covid and new cases and recovered cases. From the application user also finds out the new state wise restrictions and vaccination updates and testing history and also the covid heat map to find out where the disease has spread. From our application point of view the risk

of the data relies on the api that we are using for all these valuable data. If the api doesn't provide the accurate data then it will have a very bad impact for those who are using it. Suppose, if admin somehow forgot to update the new covid rules or if the rules and restrictions are changing frequently then it would be tough of an admin to provide the updated data regularly so as a consequence it would have misled the users and the community. So for the safety purpose we have to ensure that rules and regulations are updated regularly and the data provided by the api are correct.

References:

1. Gerald Kotonya and Ian Sommerville, "Integrating Safety Analysis and Requirements Engineering, " International Computer Science Conference, 1997.
2. Clifton A. Ericson, Hazard Analysis Techniques for System Safety, John Wiley & Sons, 2005.
3. A. b. Abdullah and Shaoying Liu, "Hazard analysis for safety-critical systems using SOFL," *2013 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, 2013, pp. 133-140, doi: 10.1109/CIES.2013.6611740.
4. Karen Allenby and Tim Kelly, "Deriving Safety Requirements using Scenarios", Fifth IEEE International Symposium on Requirements Engineering, 2001.
5. Jianwen Xiang, Kokichi Futatsugi and Yanxiang He, "Fault Tree and Formal Method in System Safety Analysis", Fourth International Conference on Computer and Information Technology, 2004.

Assignment 7

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's :

1. Md Sabbir Ahmed --1382361
2. Shrabanti Saha Rimi --1377509
3. Ashis Banik -- 1377253
4. Syed Fawzul Azim--1364224

1.1. Project Management for All groups

(a) Please finalize your Domain Model, incl. Use Cases, predominantly Use Case Texts and necessary UML diagrams, i.e. Use Case Diagrams, Collaboration Diagrams, Sequence Diagrams and Activity Charts!

(b) Build an early Design Model for the architectural design!

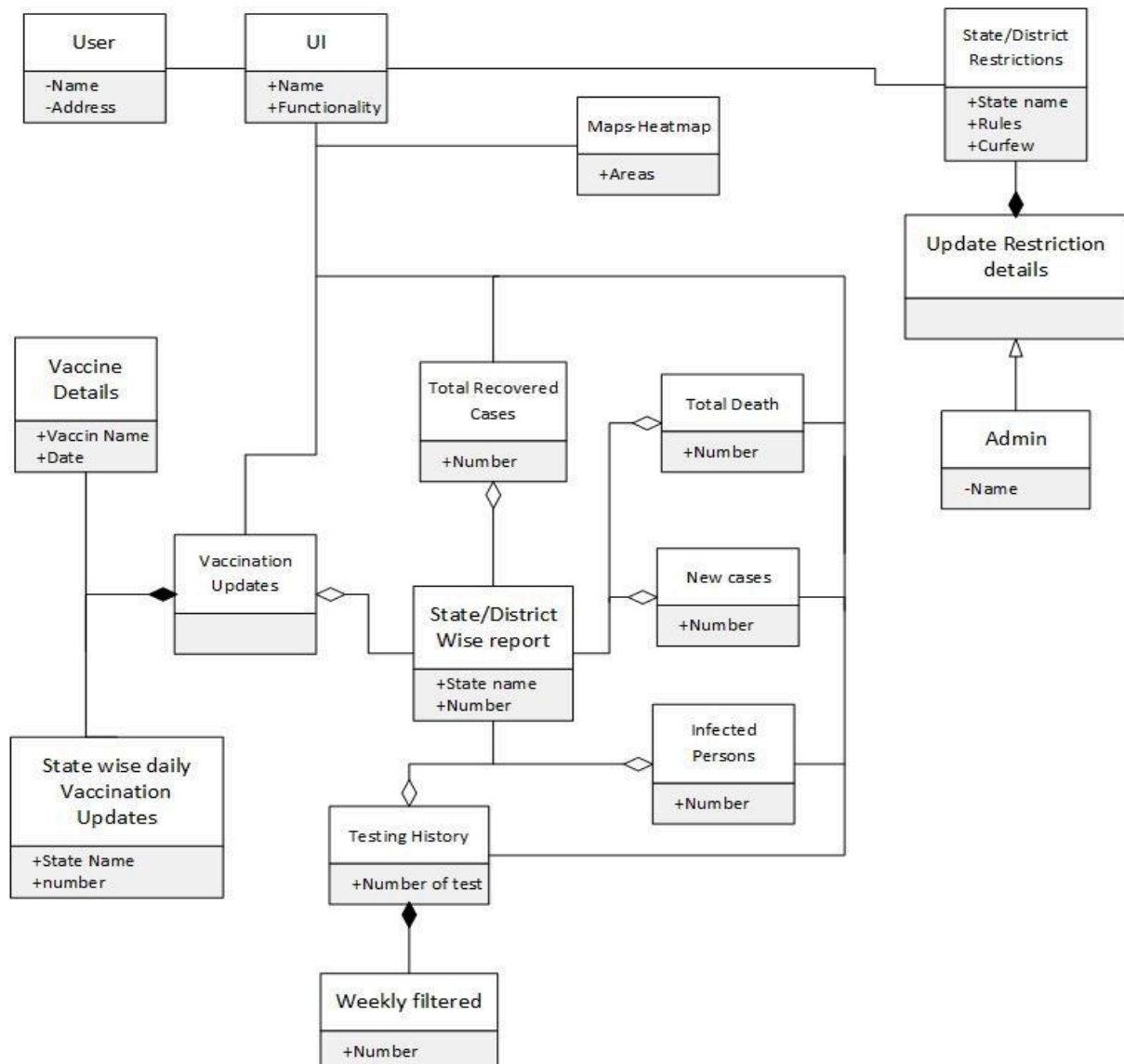
(c) Perform a classical Hazard Analysis of your design using one of the methods discussed!

(d) Establish a Safety Plan!

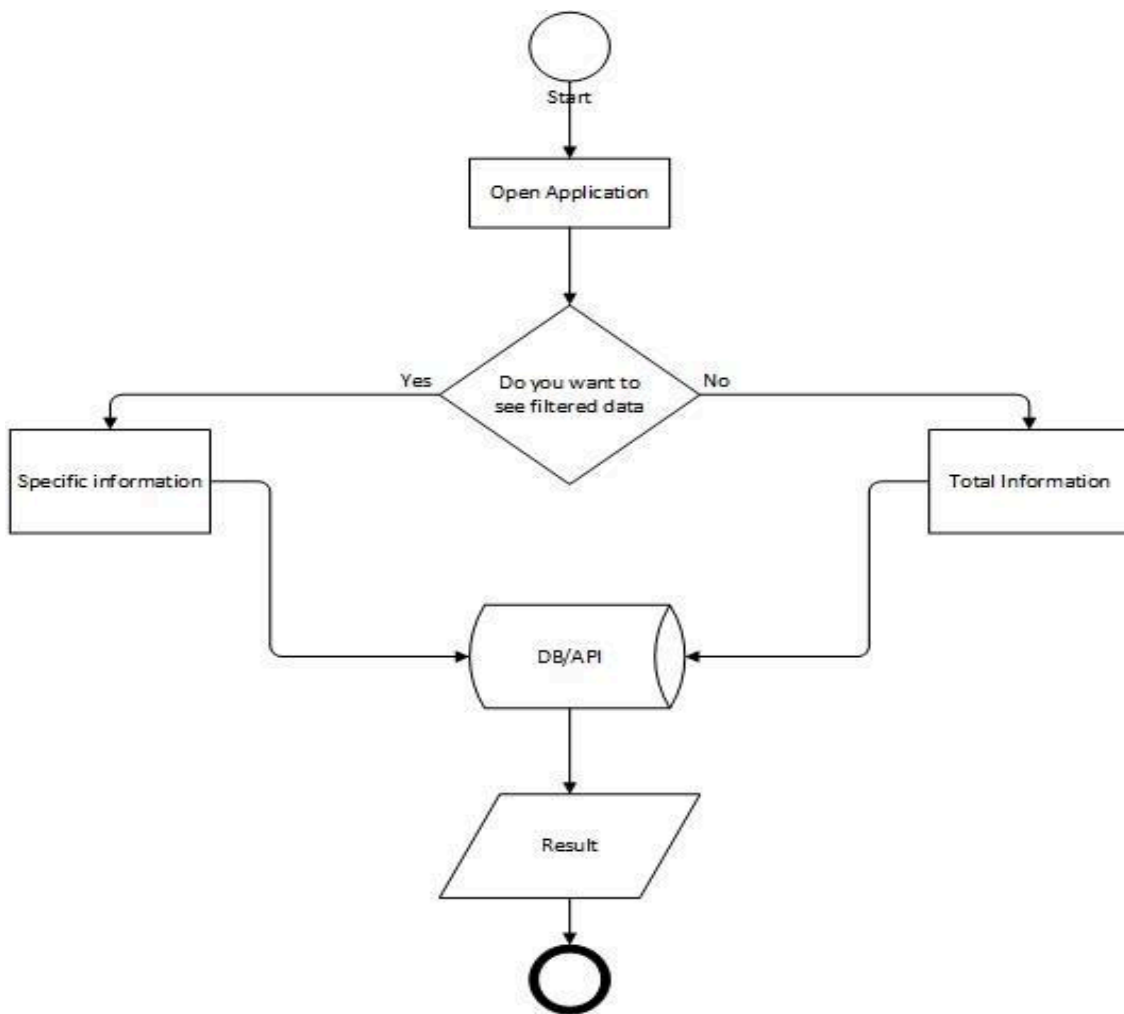
(e) Update your project plan!

(f) From the Safety Requirements in your safety plan derive Safety Constraints for the hierarchy levels and components of your system.

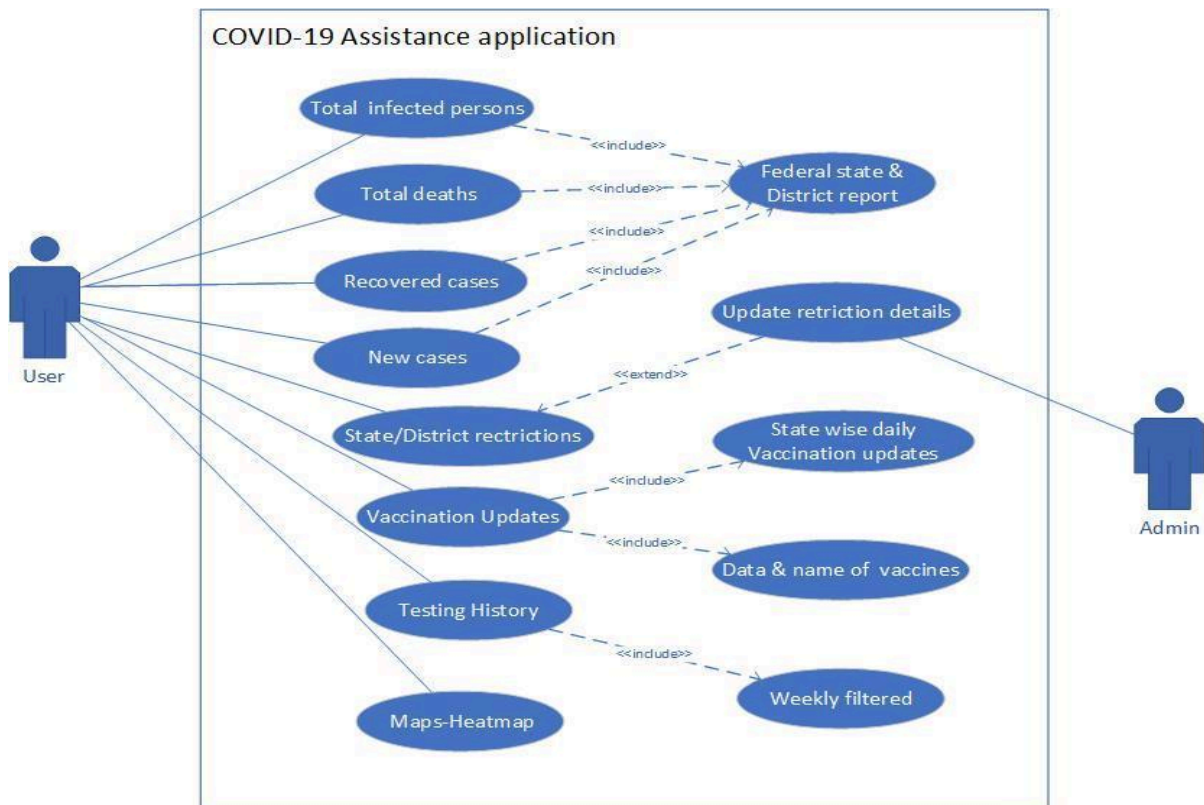
Domain Model:



Activity Chart:



Formal Use Case



Use case	Check total infections
Actors	User
Actor Intentions	Users select the 'Total Infected persons' option & see the number of infections.
Cross References	Check data from API/Database.
System Responsibility	System shows the total infected person.
Alternative Courses	If User wants to see the infection number state/district wise they can see from the 'Federal state & District report' option.

Use case	Check Total deaths, New cases, Recovered cases
Actors	User
Actor Intentions	Users can select Total deaths/New cases/Recovered cases options to see the number of Total deaths/New cases/Recovered cases.
Cross References	Check data from API/Database.

System Responsibility	1.System checks all information & response to the required field. 2.Provide information to all users synchronously.
Alternative Courses	Users can see all sections state/district wise information.

Use case	Check State/District Restrictions
Actors	User
Actor Intentions	Users can see State/District wise restrictions. e.g. which city/state call or remove lockdown, new rules & restrictions, about social distance & mask rules, party rules, curfew rules etc.
Cross References	Check from Database.
System Responsibility	Check the updated news & provide to the uses.
Alternative Courses	

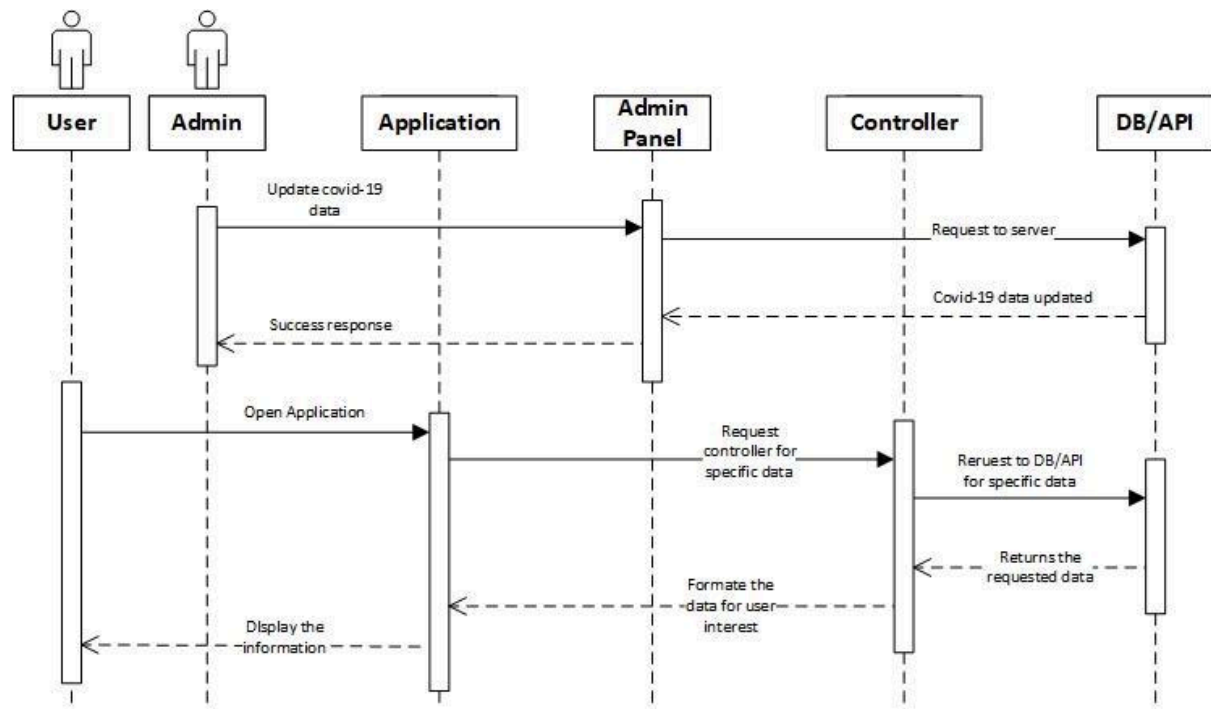
Use case	Vaccination updates & Testing history
Actors	User
Actor Intentions	Users open this field to gather knowledge about vaccination updates & testing history.
Cross References	API/Database.
System Responsibility	System verifies the Vaccination & testing data from the source and serves it.
Alternative Courses	1.User can check the date & name of vaccinations. 2.user can see weekly testing history. 3.User can check 'State wise daily Vaccination updates'.

Use case	Maps-Heatmap
Actors	User
Actor Intentions	Users can point out the intensity from the colored overlay on

	top of the map.
Cross References	API/Database.
System Responsibility	System check data from data server & show the current high & low spread areas on map.
Alternative Courses	

Use case	Update Restrictions details
Actors	Admin
Actor Intentions	Admin update all rules & regulations, restrictions, about social distance & mask rules, party rules, curfew rules etc.
Cross References	Database
System Responsibility	System updated information on database.

Sequence Diagram:



Classical Hazard Analysis (FMEA)

Failure modes and effects analysis (FMEA) is a simple procedure for systematically revealing possible failures of a structure or process as early as in the design or project stage and avoiding or mitigating them. The basic idea is that the prevention of failures is better and cheaper than their later detection and repairs.

With regard to our use case of the system, below are the points for hazard analysis:

A	B	C	D	E	F	G	H	I	J	K	L
Item No.	Function or Process Step	Failure Type/Mode	Potential Impact	SEV	Potential Causes	OCC	Detection Mode	DET	RPN	Recommended Actions	Responsibility
Item Number	Briefly outline function, step or item being analyzed	Describe what has gone wrong	What is the impact on the key output variables or internal requirements?	How severe is the effect to the User?	What causes the key input to go wrong?	How frequently is this likely to occur?	What are the existing controls that either prevent the failure from occurring or detect it should occur	How easy is it to detect?	Risk priority number (SEV*OCC*DET)	What are the actions for reducing the occurrence of the cause or improving the detection?	Who is responsible for the recommendation?
1	Accessibility of web interface/System	User does not get the updated information about Covid 19	System is not accessible	10	System access failure	4	Checking the system (Validation and verification) and other means	3	120	- Check wifi and mobile data -Check the URL (Uniform Resource Locator)	Group D
2	Authorized Access for the user's personal information should not	Unauthorized user access the personal information	Unauthorized access	9	Error by user's information	3	Checking the system (Validation and verification) and other means	4	108	-Check the personal information	Group D
3	Visualization of graph/maps	User does not get the Graph/map of accurate result of the selected countries/regions	System is not accessible	7	System access failure	3	Checking the system (Validation and verification) and other means	3	63	- Check wifi and mobile data -Check the URL (Uniform Resource Locator)	Group D
4	Vaccination updates & input	User is not able to input vaccination data	System is not accessible	8	System access failure	2	Checking the system (Validation and verification) and other means	4	64	Admin Cant give/update vaccination details	Group D
5	Update Restrictions data	Admin cant give/update restriction details	System is not accessible	9	System access failure	3	Checking the system (Validation and verification) and other means	2	54	-Should show error message if the user is unauthorized -Check the input data from user (Only give the access to authorised)	Group D

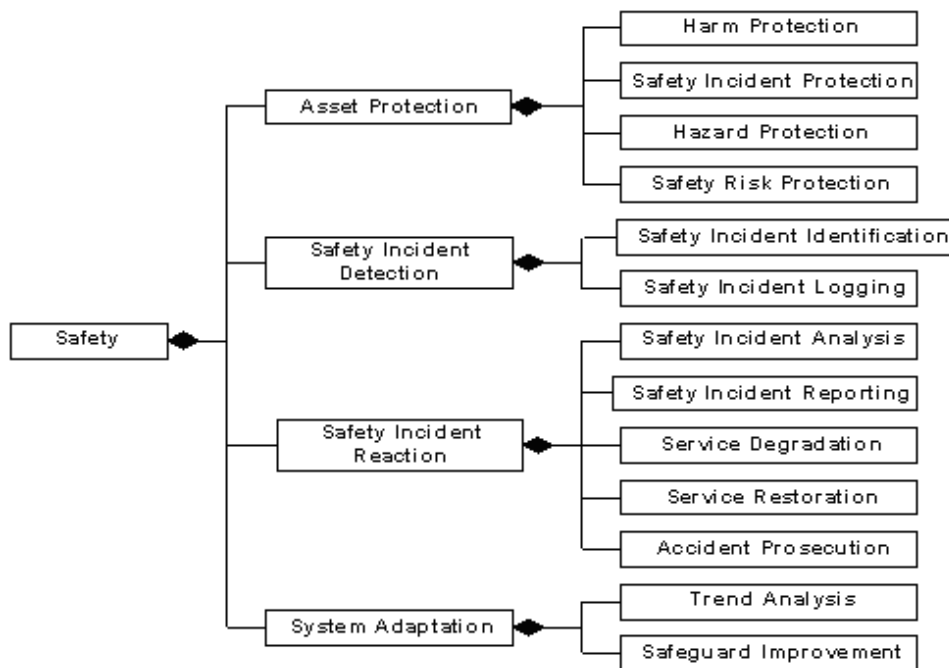
Scoring legend:

	Ranking	Low Number	High Number
Severity	1-10	Low impact	High impact
Occurrence	1-5	Lowest probability to occur	Highest probability to occur
Detection	1-5	Highest probability to detect	Lowest probability to detect

Safety

Safety is the degree to which accidental harm is properly addressed (e.g., prevented, identified, reacted to, and adapted to). Safety is classified into the following quality factors:

- 1.Health safety
- 2.Property safety
- 3.Environmental safety



Safety Requirements:

As this illustrated, there are a great many safety subfactors from which to choose. Thus, there are a great many types of safety requirements to be considered. Too often, true safety requirements are not engineered. Even when they are, the requirements engineer or safety engineer most often think in terms of specifying one of the four types of asset protection requirements. After all, it is better to prevent accidents (and near accidents) than to have to detect and react to them after they occur. Still because some safety incidents will occur no matter how much we try to engineer them out of our systems, it remains important to engineering safety incident detection and reaction requirements. Only in relatively intelligent systems are engineers beginning to think about specifying system adaptation requirements. With the preceding in mind, I will present a few safety subfactors and representative examples of their corresponding safety requirements:

Safety Subfactor	Safety Requirement
------------------	--------------------

Harm Protection	COVID-19 Assistance Application system shall not injure passengers sufficiently to require hospitalization .
Hazard Protection	COVID-19 Assistance Application shall not start moving when the results are showers more than once per week.
Safety Incident Identification	COVID-19 Assistance Application shall identify a combination of results with a probability of at least 99.99%.
Safety Incident Reporting	COVID-19 Assistance Application shall report to the safety officer occurrences of identified safety incidents at least 99.999% of the time.

SAFETY CONSTRAINTS

Rather than safety requirements, many industry and governmental standards and regulations typically concentrate on the specification of safety constraints. As defined in the following hierarchical list, safety constraints are clearly another way of specifying safety-related requirements:

- 1.A requirement is any mandatory, externally observable, verifiable (e.g., testable), and validatable behavior, characteristic, or interface.
- 2.A constraint is any engineering decision (e.g., architectural mechanism, design decision, implementation technique) that has been selected to be imposed as a requirement.
- 3.A safety constraint is any constraint that specifies a specific safeguard (e.g., architectural safety mechanism, safety design feature, safety implementation technique).

2. STAMP is a modern hazard analysis method:

- Please read at least one of the publications about STAMP!
- Summarize the ideas of STAMP in your own words!

In systems theory, complex systems are modeled as a hierarchy of levels of organization, each more complex than the one below, where a level is characterized by having emergent or irreducible properties.

Safety is an emergent property of systems that arises from the interaction of system components. Determining whether a plant is acceptably safe, for example, is not possible by examining a single valve in the plant. In fact, statements about the “safety of the valve” without information about the context in which that valve is used, are meaningless.

Conclusions can be reached, however, about the reliability of the valve, where reliability is defined as the probability that the behavior of the valve will satisfy its specification over time and under given conditions.

The hypothesis underlying the new model, called STAMP (Systems-Theoretic Accident Model and Processes) is that system theory is a useful way to analyze accidents, particularly system accidents. In STAMP, systems are viewed as interrelated components that are kept in a state of dynamic equilibrium by feedback loops of information and control. A system in this conceptualization is not a static design—it is a dynamic process that is continually adapting to achieve its ends and to react to changes in itself and its environment. The original design must not only enforce appropriate constraints on behavior to ensure safe operation, but the system must continue to operate safely as changes occur.

The basic concepts in STAMP are constraints, control loops and process models, and levels of control.

The most basic concept in the new model is not an event, but a constraint. In systems theory, control is always associated with the imposition of constraints. The cause of an accident, instead of being understood in terms of a series of events, is viewed as the result of a lack of constraints imposed on the system design and on operations, that is, by inadequate enforcement of constraints on behavior at each level of a socio-technical system. In systems theory terminology, safety is an emergent property that arises when the system components interact within an environment.

STAMP uses the concept of imposing constraints in system behavior to avoid unsafe events or conditions rather than focusing on avoiding individual component failures. Its procedure are discussed below:

1. Identify the hazard involved in the loss.
2. The hierarchical safety control structure related to the hazard is constructed and the constraints necessary to control the hazard are identified for each level.
3. Starting from the technical process and using the proximate events and general application knowledge, any failures and dysfunctional interactions involved in the loss are identified.
4. For each constraint, a determination is made about why it was violated

A new hazard analysis technique: STPA (STAMP-Based Process Analysis)

- Views safety as a dynamic control problem rather than a component failure problem
 - Accidents are the result of the inadequate inadequate control control
 - Result from lack of enforcement of safety constraints
- Prevent failures → Enforce safety constraints
- Can be used to drive the earliest design decisions
 - Can also be applied in an after-the-fact analysis and hazard assessment

Updated Project Plan

[illegible]

Assignment 08

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's :

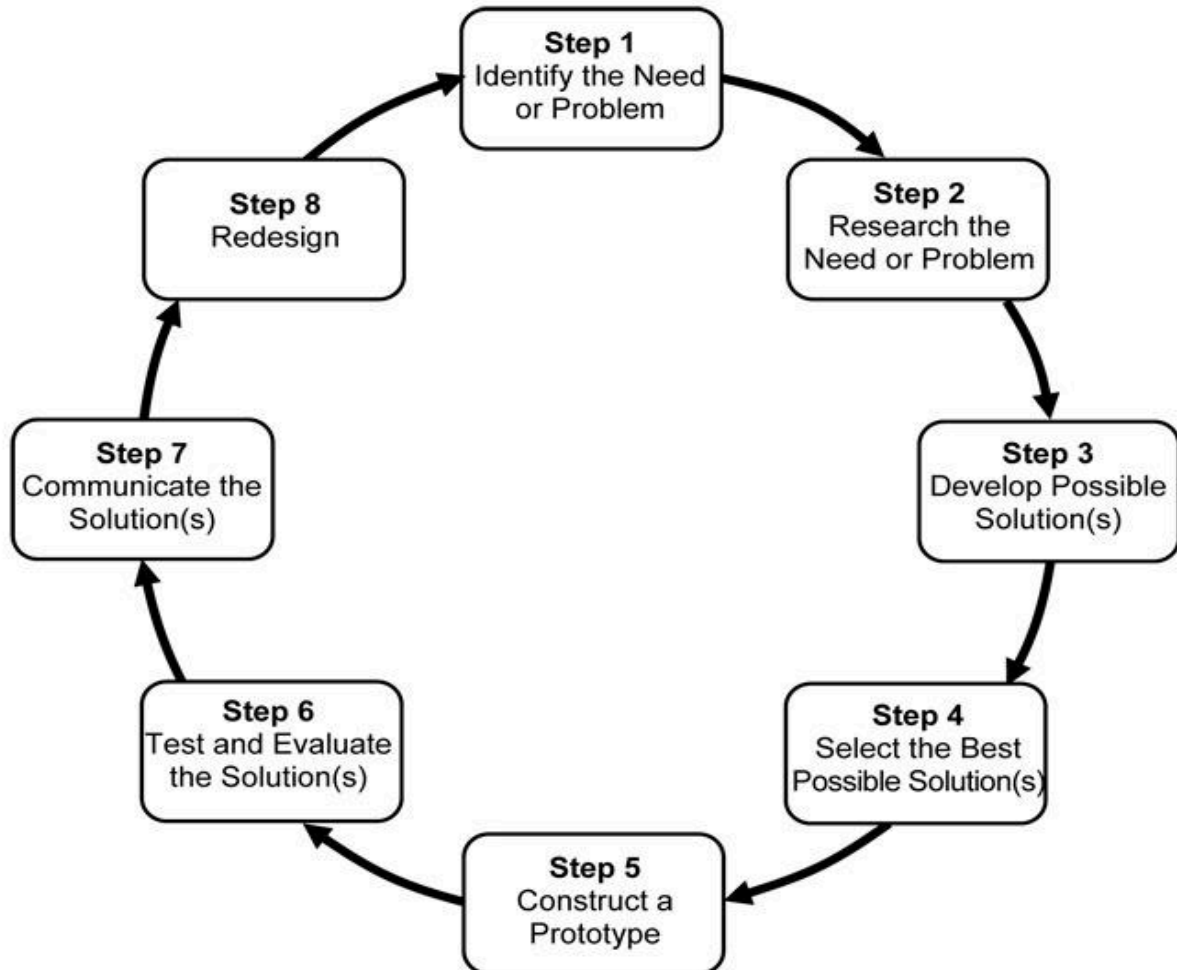
1. Md Sabbir Ahmed --1382361
2. Shrabanti Saha Rimi --1377509
3. Ashis Banik -- 1377253
4. Syed Fawzul Azim--1364224

1. Please have a look at the Balgos thesis and provide a summary!

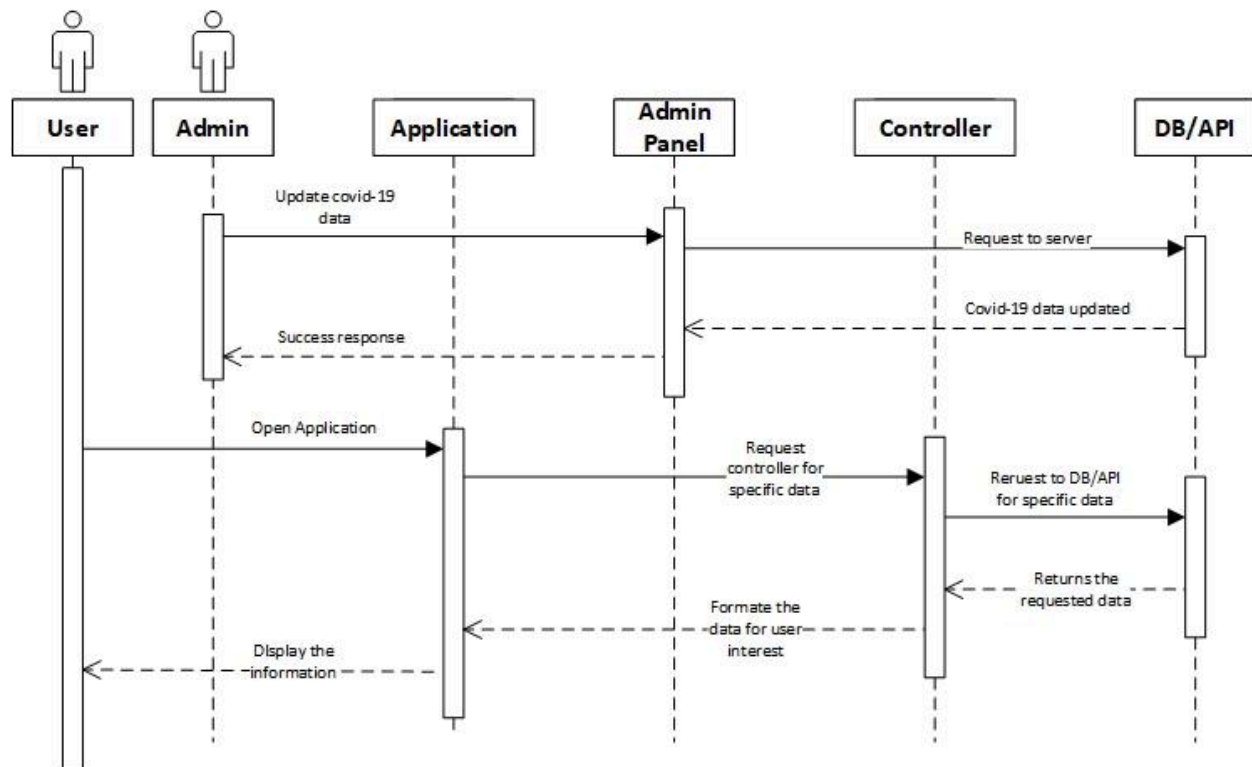
In today's environment, medical technology is rapidly advancing to deliver tremendous value to physicians, nurses, and medical staff in order to support them to ultimately serve a common goal: provide safe and effective medical care for patients. However, these complex medical systems are contributing to the increasing number of healthcare accidents each year. Specifically, the need for safe and effective diagnostic systems. With innovation in technology, come increasing concerns of maintaining system safety. Traditional, linear risk analysis methodologies recommended by the regulatory bodies may not be capable of identifying complex hazards with multiple failures, or hazards that occur sans failures. A new, systems approach to safety is needed to adapt to the increasing complexities and emerging dynamics of this technology. Based on the findings of the CAST analysis of a real-life case accident involving a medical diagnostic analyzer, the systems approach was superior to the industry standard FMECA practice in identifying hazards. It was able to detect significant contributors to the case accident in the form of failures (foreign material on the sensor), and non---failures (a conflict in controlling actions). From these identified hazards, new system safety requirements, such as establishing safer control settings, were generated to control the system from migrating to an unsafe state. This is the ultimate value that the CAST analysis can provide for the design and development of complex medical systems. This thesis confirms that the CAST and the STAMP approach was more effective in designing safety in medical diagnostic systems than the current industry standard practice of FMECA. The quantity and quality of hazards discovered with the CAST methodology are overall more productive in generating effective safety design requirements and recommendations in preventing medical accidents. A holistic approach in risk analysis can provide more value than the current linear techniques. With this system's methodology, the case accident could have been averted. Further expanding the CAST practice to other areas of medical technology development may prohibit future massive, disastrous medical accidents similar to those that gave 86 births to the FDA. Finally, the system's way will prevent history from repeating itself, and lead to new heights of safer and more effective medical care and innovation. It further confirms to the author that the system thinking is a valuable mental model and can be applied to a variety of applications in addition to safety. This is justified in the System Design and Management program, professional work experience, and in personal activities.

2. Project Work:

Design Model!



Updated Sequence Diagram



Perform a STAMP hazard analysis on the architectural level of your system! Perform a STAMP hazard analysis on at least one deeper level of your system!

STAMP uses the concept of imposing constraints in system behavior to avoid unsafe events or conditions rather than focusing on avoiding individual component failures. Its procedure are discussed below:

1. Identify the hazard involved in the loss.
2. The hierarchical safety control structure related to the hazard is constructed and the constraints necessary to control the hazard are identified for each level.
3. Starting from the technical process and using the proximate events and general application knowledge, any failures and dysfunctional interactions involved in the loss are identified.
4. For each constraint, a determination is made about why it was violated

A new hazard analysis technique: STPA (STAMP-Based Process Analysis)

- Views safety as a dynamic control problem rather than a component failure problem

- Accidents are the result of the inadequate inadequate control control
 - Result from lack of enforcement of safety constraints
 Prevent failures → Enforce safety constraints
- Can be used to drive the earliest design decisions
- Can also be applied in an after-the-fact analysis and hazard assessment

STPA (STAMP-Based Process Analysis) hazard analysis method for our system:

1. Identification of Accident, Hazards, Safety Constraints:

Accidents	Hazards	Safety Constraints
I. People get wrong covid related information and getting sick.	I. System giving wrong information and incorrect data about covid patients.	I. System must not show wrong information and data about patients.
I. Receiving wrong data about local covid patients and getting exposed to covid and dying.	II. System showing wrong heat map.	II. System must not provide an incorrect heat map.

Figure 1 : Accident, hazards, safety constraints table

2. High-Level Control Structure:

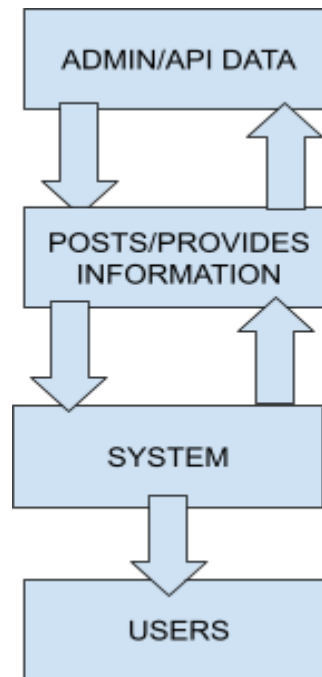


Figure 2: High level control structure

3. High-Level Control Loops:

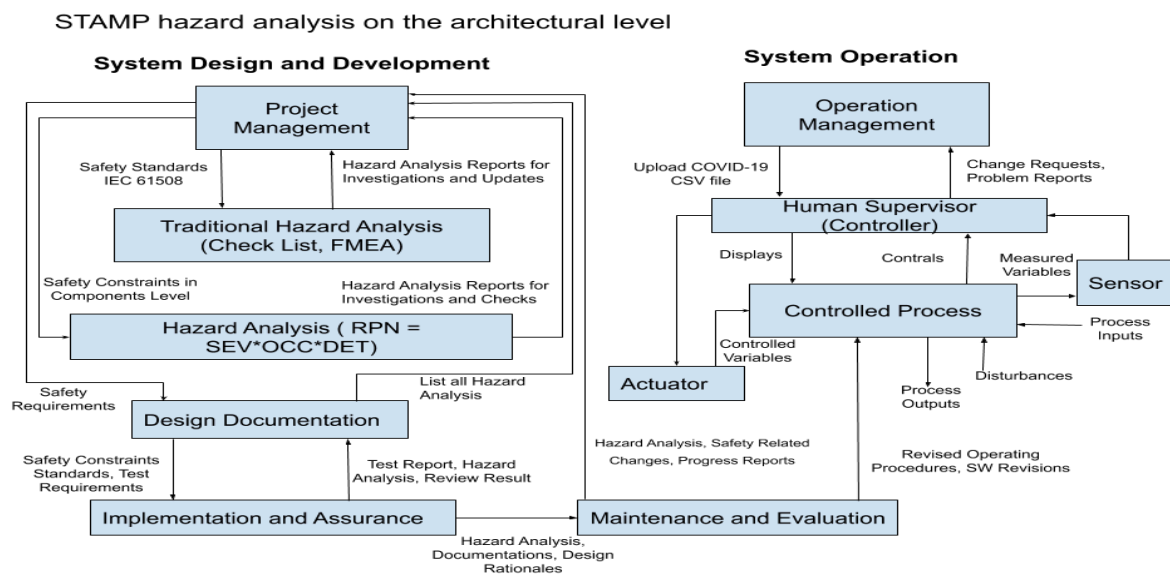


Figure 3: STAMP hazard analysis with control loops

Between the hierarchical levels of each control structure, effective communications channels are needed, both a downward *reference channel* providing the information necessary to impose constraints on the level below and an upward *measuring channel* to provide feedback about how effectively the constraints were enforced. Feedback is critical in any open system in order to provide adaptive control. At each level, inadequate control may result from missing constraints, inadequately communicated constraints, or from constraints that are not enforced correctly at a lower level.

In our System Design and Development, Project Management follows safety standards while analyzing hazards of the system (in our system we prepare a hazard checklist and make FMEA based on the checklist) and get the reports. Risk Priority Number was calculated following safety constraints in components level and got the feedback report after investigations and checking. The impact of specific decisions at each level on the objectives and values passed down are adequately and formally evaluated. At the end of the development process, the results of the hazard analyses as well as documentation of the safety-related design features and design rationale passed on to the maintenance group to be used in the change process.

4. Control flow leading to hazard:

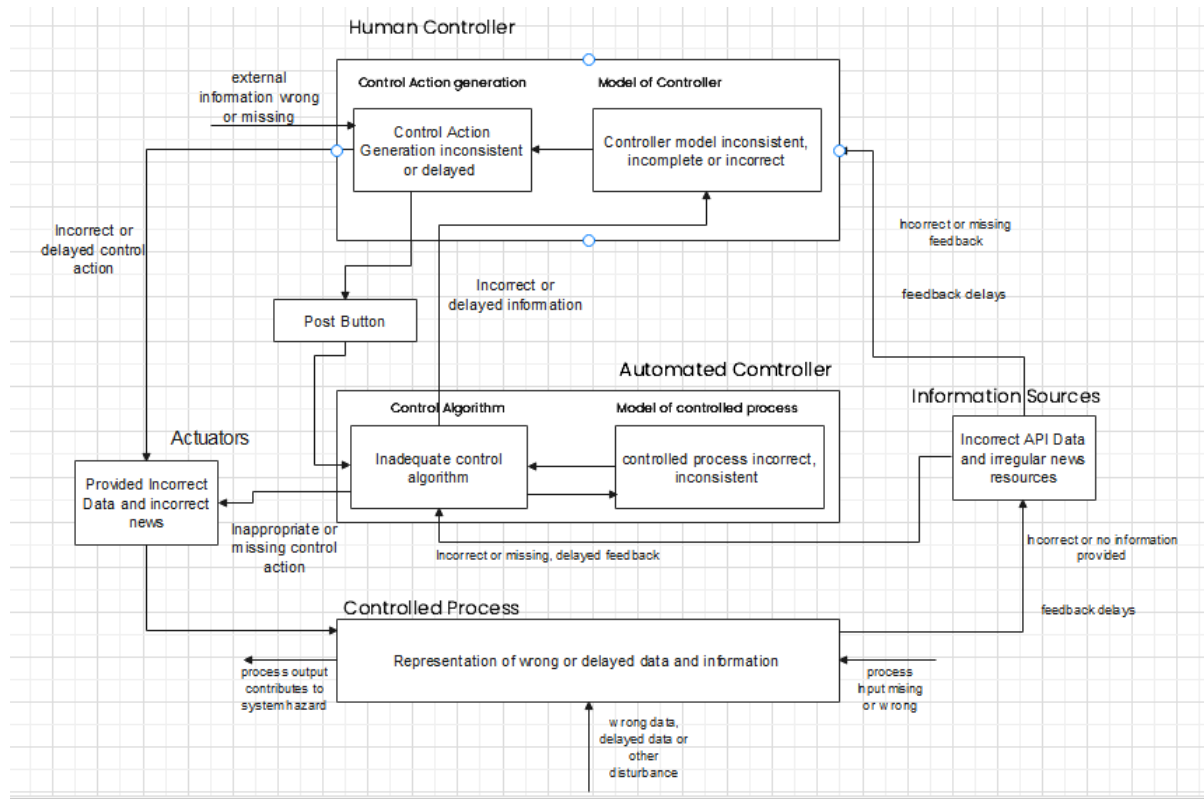


Figure 4: Detailed control flow leading to hazard

4. Tesla

Tesla Summon accident:

Tesla:

- "the incident occurred as a result of the driver not being properly attentive..."
- Drivers must agree to legal terms on their touch screen before the feature is allowed

As per Tesla , This feature will park Model S while the driver is outside the vehicle. Please note that the vehicle may not detect certain obstacles, including those that are very narrow (e.g., bikes), lower than the fascia, or hanging from the ceiling. As such, Summon requires that you continually monitor your vehicle's movement and surroundings while it is in progress and that you remain prepared to stop the vehicle at any time using your key fob or mobile app or by pressing any door handle. You must maintain control and responsibility for your vehicle when using this feature and should only use it on private property."

For the tesla summon feature to work the human factor of the system must always perform without fail and human factor of the system is the most critical of all. Tesla warns owners to be careful with using Smart Summon because it's not a fully autonomous feature. "You are still responsible for your car and must monitor it and its surroundings at all times and be within your line of sight because it may not detect all obstacles," the fine print on Tesla's website reads. "Be especially careful around quick moving people, bicycles and cars."

Reference

[1]<https://www.cnbc.com/2019/10/02/nhtsa-looking-into-tesla-accidents-with-smart-summon-feature.html>

Assignment 9

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

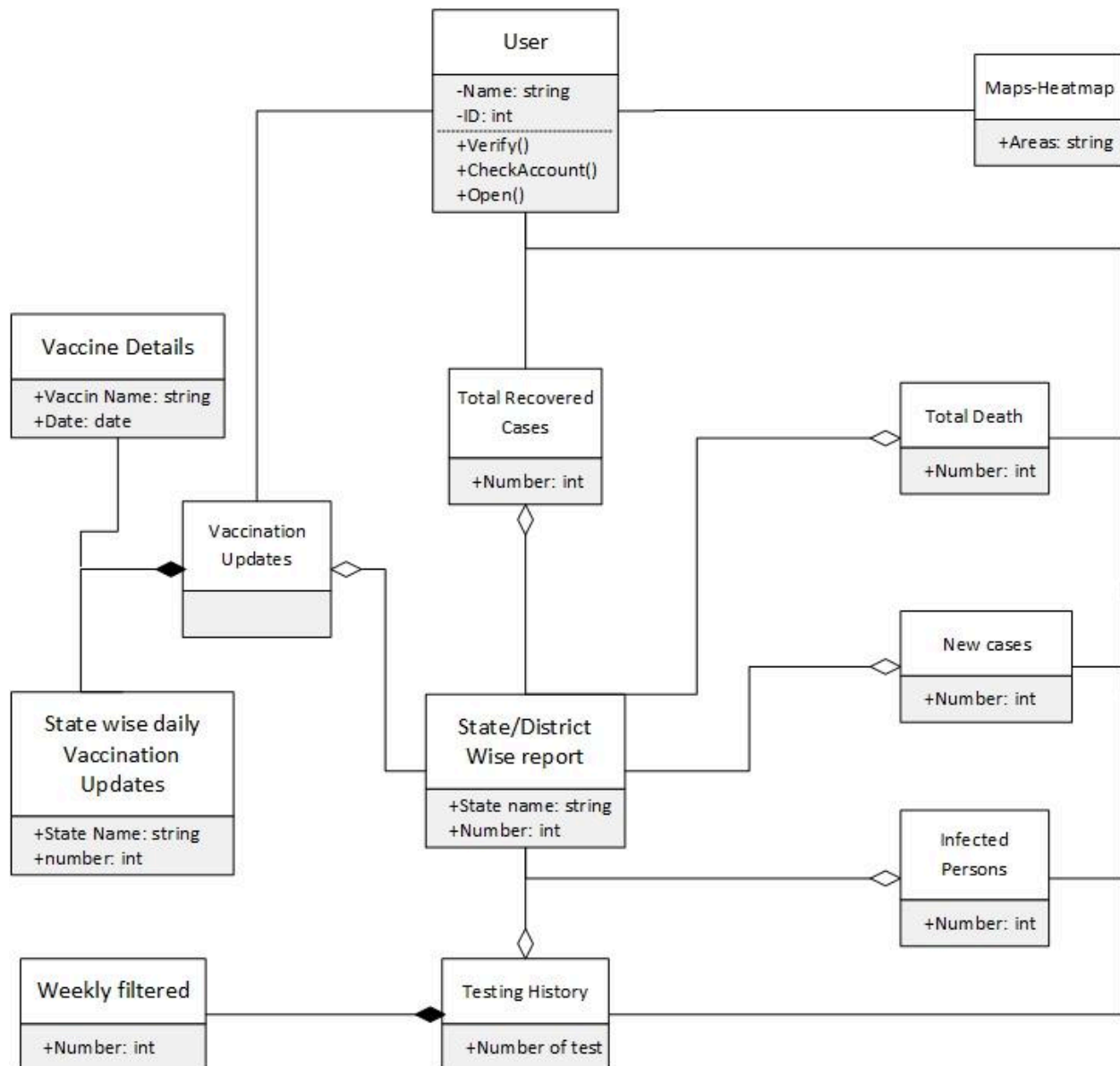
1. Md Sabbir Ahmed --1382361
2. Shrabanti Saha Rimi-- 1377509
3. Ashis Banik -- 1377253
4. Syed Fawzul Azim-- 1364224

1. Transition from Design to Implementation:

- Enhance your Design Model in detail!
- Use all UML diagram types necessary!
- Test your data analysis algorithms in detail!

1. Transition from Design to Implementation:

Class Diagram



linear regression algorithm: We want to use linear regression algorithm for our data analysis. It will be useful for data analysis in our project.

Linear regression is one of the most basic algorithms of advanced analytics. This also makes it one of the most widely used. People can easily visualize how it is working and how the input data is related to the output data.

Linear regression uses the relationship between two sets of continuous quantitative measures. The first set is called the *predictor* or *independent variable*. The other is the *response* or *dependent variable*. The goal of linear regression is to identify the relationship in the form of a formula that describes the dependent variable in terms of the independent variable. Once this relationship is quantified, the dependent variable can be predicted for any instance of an independent variable.

Linear regression is commonly used for predictive analysis and modeling. For example, it can be used to quantify the relative impacts of age, gender, and diet (the predictor variables) on height (the outcome variable).

02. Please review literature about the Automation Hype and its problems!

What is Automation:

Automation is the application of technology, programs, robotics or processes to achieve outcomes with minimal human input.

Automation is a broad term that can cover many areas of technology where human input is minimized. This can include everything from business-specific types such as: business process automation (BPA), IT automation, marketing automation and industrial automation. It also covers personal applications such as home automation.

Types of Automation

- Mechanical Automation (eg Factory > Industrial Revolution)
- Mixed ; Information and Mechanical (eg Aircraft)

1. Decision Aid
2. Supervisory Control

· Supervisory Control:

1. Autopilot - Flight Management System
2. Autonomous Vehicle
3. Process Control Plant
4. Thermostat
5. Word Processing Program
6. Cruise Control

· Decision Aid

1. Alerting Systems :Gear Warning ,Idiot Light (eg Oil Pressure) , TCAS
2. Automated Planning Systems : Path Planners
3. Suggesters :Spell Check
4. Data Analysis : Filters ,Interactive Data Analysis Tools

Automation strength:

Worker safety is an important reason for automating an industrial operation. Automated systems often remove workers from the workplace, thus safeguarding them against the hazards of the factory environment. In the United States **the Occupational Safety and Health Act of 1970** (OSHA) was enacted with the national objective of making work safer and protecting the physical well-being of the worker. OSHA has had the effect of promoting the use of automation and robotics in the factory.

- Can be fast
- Does not get bored
- Consistent
- Good for predictable cases
- Performance : Speed, Accuracy, Strength
- Enhance Human : Extend, Relieve, Backup Replace

Automation Limits :

Ø Worker displacement: Automated systems often remove workers from the workplace in the industrial sector. Despite the social benefits that might result from retraining displaced workers for other jobs, in almost all cases the worker whose job has been taken over by a machine undergoes a period of emotional stress.

Ø Cost: An automated system can cost millions of dollars to design, fabricate, and install. A higher level of maintenance needed than with a manually operated machine, and a generally lower degree of flexibility in terms of the possible products as compared with a manual system.

Ø Privacy Maintenance: The possibility that workers will become slaves to automated machines, that the privacy of humans will be invaded by vast computer data networks, that human error in the management of technology will somehow endanger civilization, and that society will become dependent on automation for its economic well-being.

Ø Adaptability

Ø Input requirements

Ø Interface with system

4. Please update your STAMP/STPA analysis!

STAMP uses the concept of imposing constraints in system behavior to avoid unsafe events or conditions rather than focusing on avoiding individual component failures. Its procedure are discussed below:

1. Identify the hazard involved in the loss.
2. The hierarchical safety control structure related to the hazard is constructed and the constraints necessary to control the hazard are identified for each level.
3. Starting from the technical process and using the proximate events and general application knowledge, any failures and dysfunctional interactions involved in the loss are identified.
4. For each constraint, a determination is made about why it was violated

A new hazard analysis technique: STPA (STAMP-Based Process Analysis)

- Views safety as a dynamic control problem rather than a component failure problem

- Accidents are the result of the inadequate inadequate control control
 - Result from lack of enforcement of safety constraints
 Prevent failures → Enforce safety constraints
- Can be used to drive the earliest design decisions
- Can also be applied in an after-the-fact analysis and hazard assessment

STPA (STAMP-Based Process Analysis) hazard analysis method for our system:

1. Identification of Accident, Hazards, Safety Constraints:

Accidents	Hazards	Safety Constraints
I. People get wrong covid related information and get sick.	I. System giving wrong information and incorrect data about covid patients.	I. All the covid related information and suggestions must be verified by an admin.
I. Receiving wrong data about local covid patients and getting exposed to covid and dying.	II. System showing wrong heat map.	II. The data for the creation of a heat map will be collected from a well trusted source.

Figure 1 : Accident, hazards, safety constraints table

2. High-Level Control Structure:

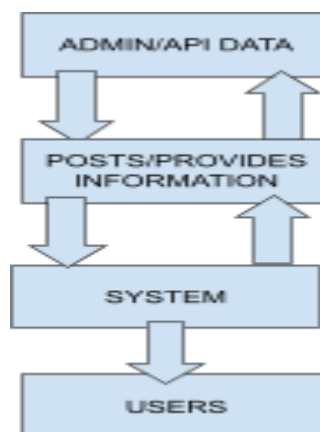


Figure 2: High level control structure

3. High-Level Control Loops:

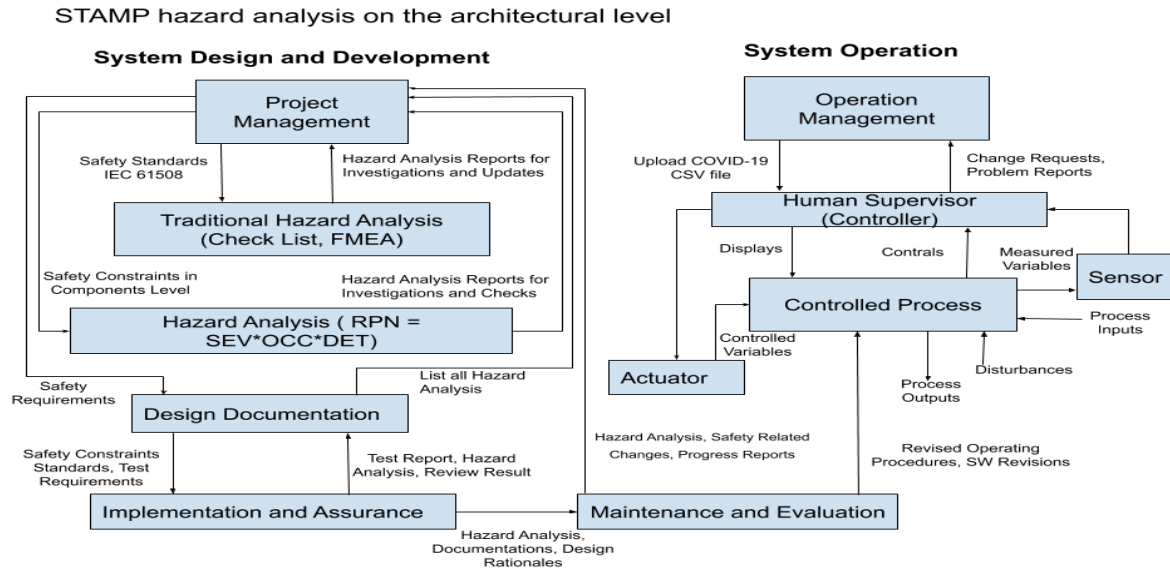


Figure 3: STAMP hazard analysis with control loops

Between the hierarchical levels of each control structure, effective communications channels are needed, both a downward *reference channel* providing the information necessary to impose constraints on the level below and an upward *measuring channel* to provide feedback about how effectively the constraints were enforced. Feedback is critical in any open system in order to provide adaptive control. At each level, inadequate control may result from missing constraints, inadequately communicated constraints, or from constraints that are not enforced correctly at a lower level.

In our System Design and Development, Project Management follows safety standards while analyzing hazards of the system (in our system we prepare a hazard checklist and make FMEA based on the checklist) and get the reports. Risk Priority Number was calculated following safety constraints in components level and got the feedback report after investigations and checking. The impact of specific decisions at each level on the objectives and values passed down are adequately and formally evaluated. At the end of the development process, the results of the hazard analyses as well as documentation of the safety-related design features and design rationale passed on to the maintenance group to be used in the change process.

4. Control flow leading to hazard:

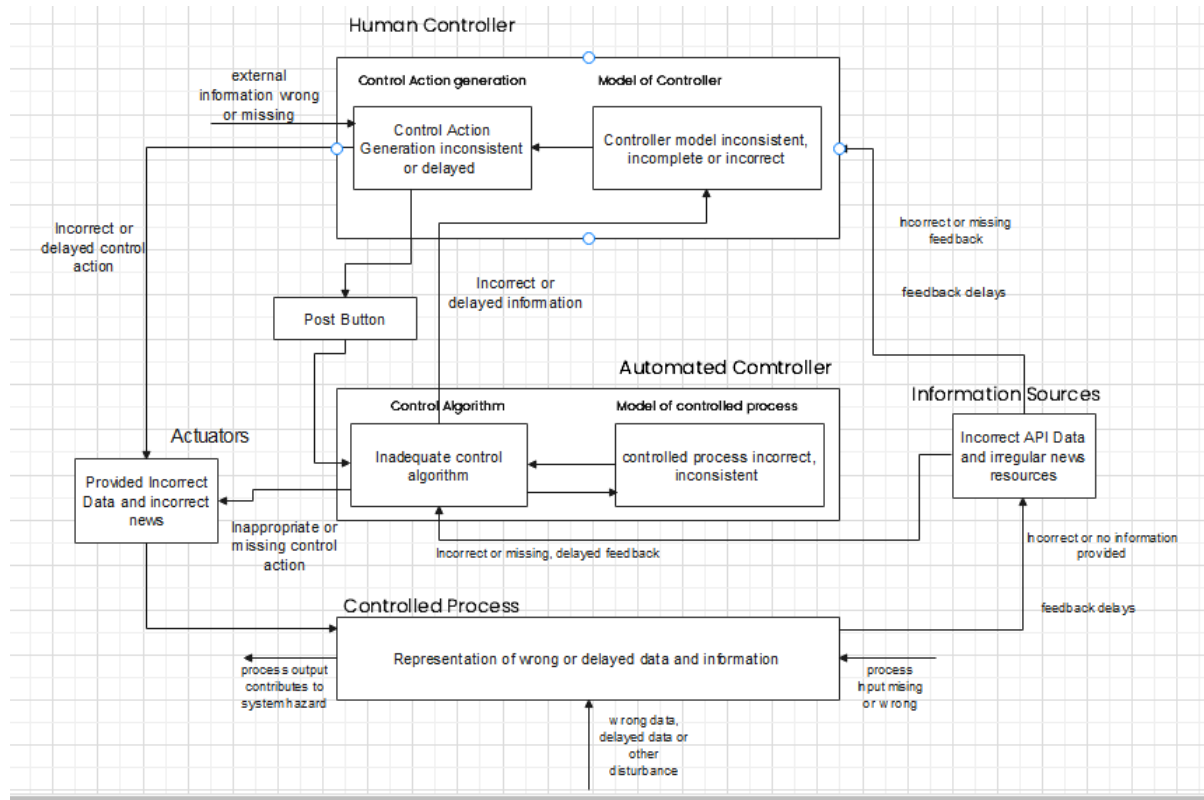










Figure 4: Detailed control flow leading to hazard


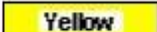

5. Please report your advances in the project in a written document:

- Status report
- Updated software plan (including estimation without display of calculation)!
- Hazard analysis concerning Safety and Security!!
- Updated Safety plan
- Security plan!

Project Status Report

For period:	July 2021
Submitted by:	Group-D
Project Name:	COVID-19 Assistance Application

Status Item	Current Status	Prior Status
Overall project status		
Project Risk		
Schedule		
Budget		

Color Key	
	Strong probability item will meet dates and acceptable quality.
	Good probability item will meet dates and acceptable quality. Schedule, resource, or scope changes may be needed.
	Probable that item will NOT meet dates with acceptable quality without changes to schedule, resources, and/or scope.

Key Upcoming Milestones:	
Prepare design model to code	17.06.2021-22.06.2021
Prepare test plan	23.06.2021-24.06.2021
Meeting to review test plan	25.06.2021-26.06.2021
Perform testing	27.06.2021-28.06.2021
Risk exposure table development	29.06.2021
Complete draft SRS	1st week of july

Complete final version of SRS	1st week of july
Send final version of SRS	1st week of july

**Updated software estimation calculation:
(Function Point Analysis,COCOMO,COCOMO II)**

Components of Function Point Analysis

Function Point Analysis Part-1:

Determine the following components:

Types of FP Attributes

Measurements Parameters	Examples
1.Number of External Inputs (EI)	Input screen and tables
2. Number of External Output (EO)	Output screens and reports
3. Number of external inquiries (EQ)	Prompts and interrupts.
4. Number of internal files (ILF)	Databases and directories
5. Number of external interfaces (EIF)	Shared databases and shared routines.

Analysis of the software system as presented in the user point of view.

The number of various components:

- 1.External Inputs (EI): 0
- 2.External Outputs (EO): 3
- 3.Inquiries (EQ): 8
- 4.Internal Logic File (ILF): 4
- 5.External Logic File (ELF): 8

The degree of complexity (Simple, Average, Complex) was evaluated for each component.

Function Point Analysis Part-2:

Compute the unadjusted function point (UFP)

- 1.rate each component as low average or high.
- 2.for transactions (EI, EO, EQ), the rating is based on the FTR, and DET. [FTR- The number of files updated or referenced, DET- The number of user recognizable fields]

Number of user recognizable fields based on the table below an EI that references 0 files and 10 user recognizable fields would be ranked as Low.

File Type Referenced(FTR)	Data Element Type(DET)		
	1-4	5-15	15+
0-1	L	L	A
2	L	A	H
3+	A	H	H

For files (ILF and ELF), the rating is based on the RET and DET.

RET- The number of users-recognizable data elements in an ILF or ELF.

DET- The number of users-recognizable fields.

Based on the table below an ILF that contains 4 data elements and 6 user recognizable fields would be ranked as Low.

Record Element Type(RET)	Data Element Type(DET)		
	1-19	20-51	51+
1	L	L	A
2-5	L	A	H
6+	A	H	H

Convert ratings into UFP's (Unadjusted Function Points)

1. Number of external inputs (EI)	3	4	6
2. Number of external outputs (EO)	4	5	6
3. Number of external inquiries (EQ)	3	4	5
4. Number of internal files (ILF)	7	10	15
5. Number of external interfaces (EIF)	5	7	10
1. Number of external inputs (EI)	3	4	6

Program Characteristic	Function Points		
	Low Complexity	Medium Complexity	High Complexity
External Inputs	$0 \times 3 = 0$	4	6
External Outputs	$3 \times 4 = 12$	5	7
External Queries	$3 \times 3 = 9$	4	6
Internal Logical Files	$7 \times 7 = 49$	10	15
External Interface Files	$5 \times 5 = 25$	7	10

Unadjusted Function Point total			95
--	--	--	-----------

Function Point Analysis Part-3:

Compute Value Adjustment Factor (VAF) based on 14 general system characteristics (GSC).

Weight each GSC on a scale of 0 to 5 based on whether it has no influence to strong influence.

- Data communications-3
- Distributed data processing-0
- Performance-0
- Heavily used configuration-2
- Transaction rate-3
- On-Line data entry-0
- End-user efficiency-3
- On-Line update-3
- Complex processing-4
- Reusability-4
- Installation ease-1
- Operational ease-3
- Facilitate change-2
- Multiple sites-5

Total VAF-33

Compute the FP as follows:

$VAF = \text{Sum (GSC)}$

$FP = UFP * (0.65 + (VAF * 0.01))$

$FP = 95 * (0.65 + (33 * 0.01)) = 93.1$

FP=93

Convert FP to line of source code (SLOC)

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
Brio +	14	14	13	16
C *	97	99	39	333
C++ *	50	53	25	80
C# *	54	59	29	70

$95UFP * 25(C++)SLOC/UFP = 2375SLOC = 2KLOC$

COCOMO

Basic COCOMO Model:

It estimates the software roughly and quickly. It is mostly for small – medium sized software.

ORGANIC:

$$\begin{aligned}\text{Effort} &= a(\text{KLOC})^b \text{ Person-month} \\ &= 2.4(2)^{1.05} \text{ Person-month} \\ &= 5 \text{ Person-month}\end{aligned}$$

$$\begin{aligned}\text{Development-Time} &= c(\text{Effort})^d \text{ Months} \\ &= 2.5(4)^{0.38} \text{ Months} \\ &= 4 \text{ Months}\end{aligned}$$

COCOMO II

Effort Equation,

$$\text{Effort} = 2.94 * \text{EAF} * (\text{KSLOC})^E$$

$$\text{Effort Adjustment Factor} = \text{EAF} = 1.09$$

$$\text{Effort} = 2.94 * (1.09) * (2)^{1.0997} = 5 \text{ Person-Months}$$

Schedule Equation,

$$\text{Duration} = 3.67 * (\text{Effort})^{SE}$$

$$\text{Duration} = 3.67 * (5)^{0.3179} = 4 \text{ months}$$



$$\text{Average staffing} = \text{Effort} / \text{Duration}$$









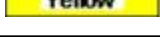
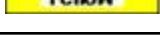

$$\text{Average staffing} = (5 \text{ Person-Months}) / (4 \text{ Months}) = 1.25 \text{ people}$$

Safety plan

Safety is the degree to which accidental harm is properly addressed (e.g., prevented, identified, reacted to, and adapted to). Safety is classified into the following quality factors:

1. Health safety
2. Property safety
3. Environmental safety

Harm Protection	
Safety Incident Protection	

Hazard Protection	
Safety Risk Protection	
Safety Incident Identification	
Safety Incident Logging	
Safety Incident Analysis	
Safety Incident Reporting	
Service Degradation	
Service restoration	
Accident Prosecution	
Trend Analysis	
Safeguard Improvement	

Hazard analysis concerning Safety and Security

Classical Hazard Analysis (FMEA)

Failure modes and effects analysis (FMEA) is a simple procedure for systematically revealing possible failures of a structure or process as early as in the design or project stage and avoiding or mitigating them. The basic idea is that the prevention of failures is better and cheaper than their later detection and repairs.

With regard to our use case of the system, below are the points for hazard analysis:

Item Number	Function or Process Step	Failure Type/Mode	Potential Impact	SEV	Potential Causes	OCC	Detection Mode	DET	RPN	Recommended Actions	Responsibility
Item Number	Briefly outline function, step or item being analyzed	Describe what has gone wrong	What is the impact on the key output variables or internal requirements?	How severe is the effect on the User?	What causes the key input to go wrong?	How frequently is this likely to occur?	What are the existing controls that either prevent the failure from occurring or detect it should occur	How easy is it to detect?	Risk priority number (SEV*OCC*DET)	What are the actions for reducing the occurrence of the cause or improving the detection?	Who is responsible for the recommended action?
1	Accessibility of web interface/System to user	User does not get the updated information about Covid 19	System is not accessible	10	System access failure	4	Checking the system (Validation and verification) and other means	3	120	Check wifi and mobile data -Check the URL (Uniform Resource Locator)	Group D
2	Authorized Access for the administrator in the system	Unauthorized user is able to access the system	Unauthorized access to the system	9	Error by controller	3	Checking the system (Validation and verification) and other means	4	108	-Should show error message if the user is unauthorized -Check the input data from user	Group D
3	The user's personal information should not be visible to all.	User's personal information is	User privacy will be hampered	8	user information security	4	Checking the system (Validation and	3	96	-Check that the personal information is	Group D

		visible to public					verification) and other means			accessible to others	
4	Visualization of graph/maps	User does not get the Graph/map of accurate results of the selected countries/regions COVID-19 information.	System is not accessible	7	System access failure	3	Checking the system (Validation and verification) and other means	3	63	- Check wifi and mobile data -Check the URL (Uniform Resource Locator)	Group D
5	Vaccination updates & Testing history	not able to input vaccination data and history	User is not able to input vaccination data and history	8	System access failure	2	Checking the system (Validation and verification) and other means	4	64	Admin can't give/update restriction details	Group D

6	Update Restrictions details	Admin can't give/update restriction details	System is not accessible	9	System access failure	3	Checking the system (Validation and verification) and other means	2	54	-Should show error message if the user is unauthorized -Check the input data from user (Only give the access to authorised user access), Otherwise throw invalid user popup.	Group D
---	-----------------------------	---	--------------------------	---	-----------------------	---	---	---	-----------	---	---------

	Ranking	Low Number	High Number
Severity	1-10	Low impact	High impact
Occurrence	1-5	Lowest probability to occur	Highest probability to occur
Detection	1-5	Highest probability to detect	Lowest probability to detect

Assignment 11

HIS/Basys - Safety Critical Computer Systems

Summer Semester 2021

Prof. Dr. Matthias F. Wagner

Member's List:

1. Md Sabbir Ahmed --1382361
2. Shrabanti Saha Rimi-- 1377509
3. Ashis Banik -- 1377253
4. Syed Fawzul Azim-- 1364224

1.

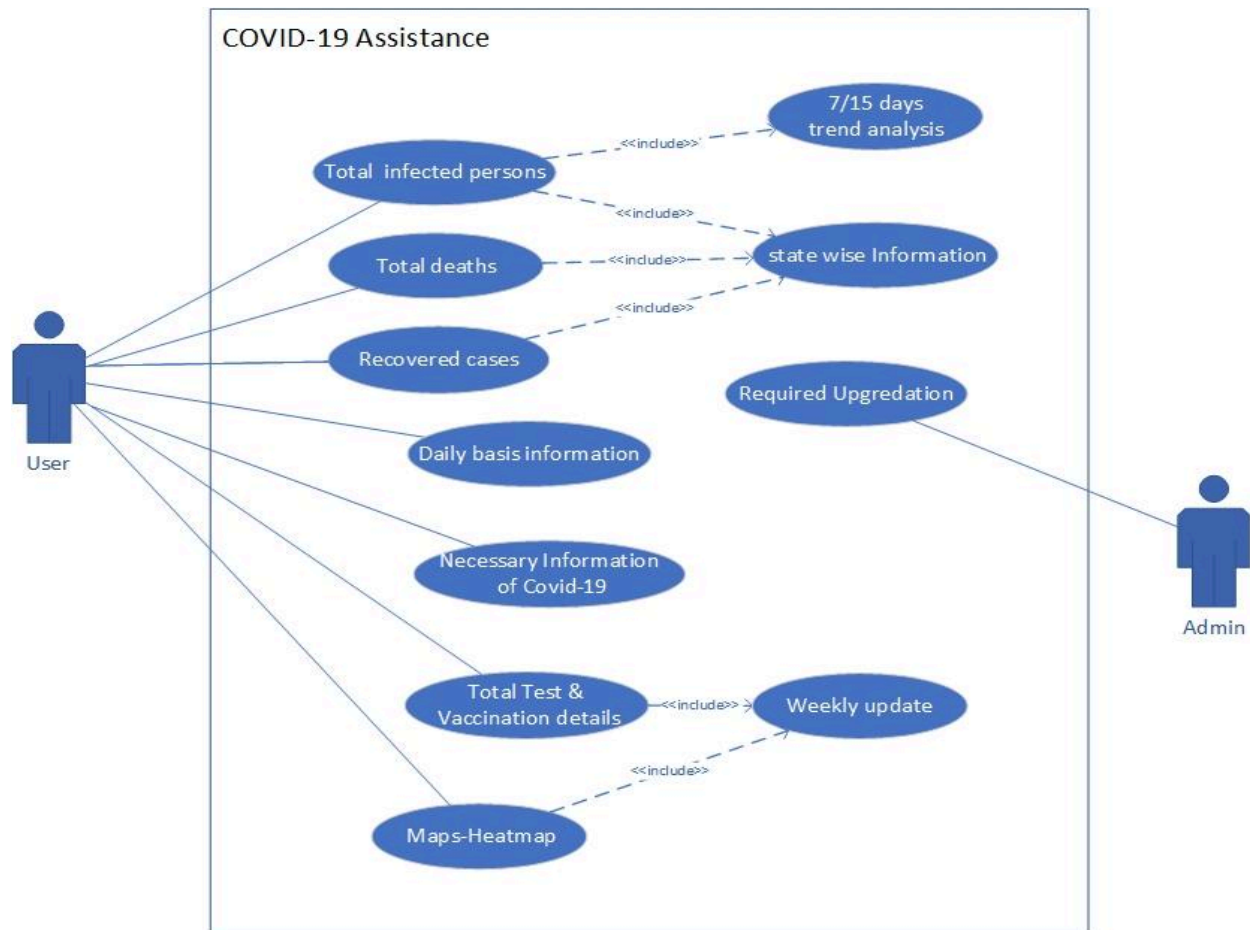


Figure-1: Use Case Diagram

Use Case Document:

Use case	Check total infected persons
Actors	User
Actor Intentions	Users select the 'Total Infected persons' option & see the number of infections.
Cross References	Check data from the system.
System Responsibility	System shows the total infected person.
Alternative Courses	If User wants to see the infection number state/district wise they can see from the 'Federal state & District report' option.

Use case	Check trend analysis
Actors	User
Actor Intentions	Users can analyze the seven or fifteen days infection rate & also be able to see the number of infections.
Cross References	Check data from the system.
System Responsibility	System shows the total infected person.

Use case	Check Total deaths, Recovered cases
Actors	User
Actor Intentions	Users can select Total deaths/Recovered cases options to see the number of Total deaths/New cases/Recovered cases.
Cross References	Check data from API/Database
System Responsibility	1.System checks all information & response to the required field. 2.Provide information to all users synchronously.

Use case	Total Test & Vaccination details
Actors	User
Actor Intentions	Users open this field to gather knowledge about vaccination updates & testing history.
Cross References	API/Database.
System Responsibility	System verifies the Vaccination & testing data from the source and serves it.
Alternative Courses	1.User can check the Vaccination data weekly. 2.user can see weekly testing history.

Use case	Daily basis Information
Actors	User
Actor Intentions	Users open this field to gather knowledge about daily new infection,recovered,deaths cases.
Cross References	API/Database.
System Responsibility	System verifies the Vaccination & testing data from the source and serves it.

Use case	Necessary Information of Covid-19
Actors	User
Actor Intentions	Users open this field to gather knowledge about Covid-19.
Cross References	API/Database.

Use case	Maps-Heatmap
Actors	User
Actor Intentions	Users can point out the intensity from the colored overlay on top of the map.
Cross References	API/Database.
System Responsibility	System check data from data server & show the current high & low spread areas on map.
Alternative Courses	1.User can check weekly updated incidents on the map.

Use case	Required Upgradation
Actors	User
Actor Intentions	Admin can update any necessary system/coding upgradation.
System Responsibility	Verify the update of the system.

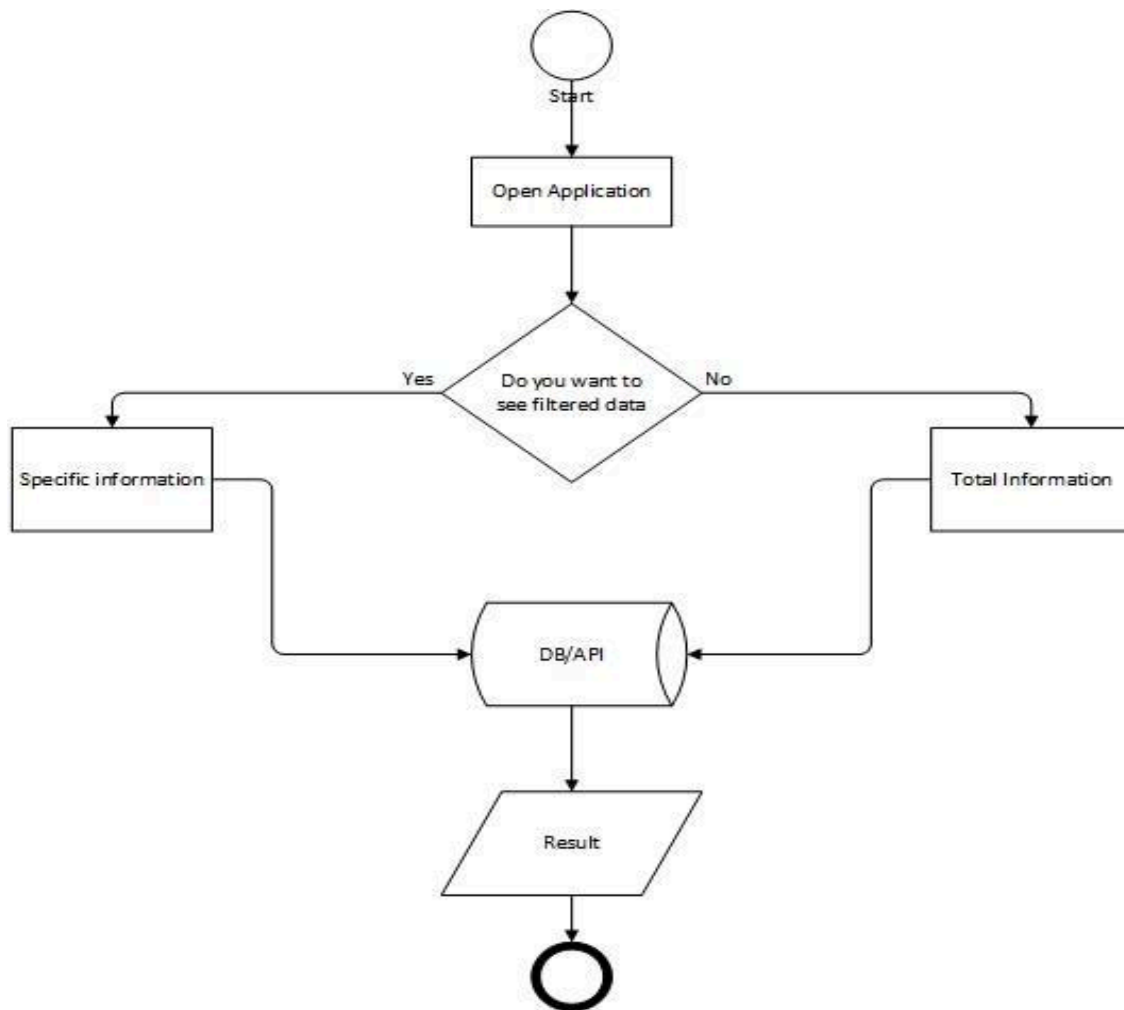


Figure-2: Activity Diagram

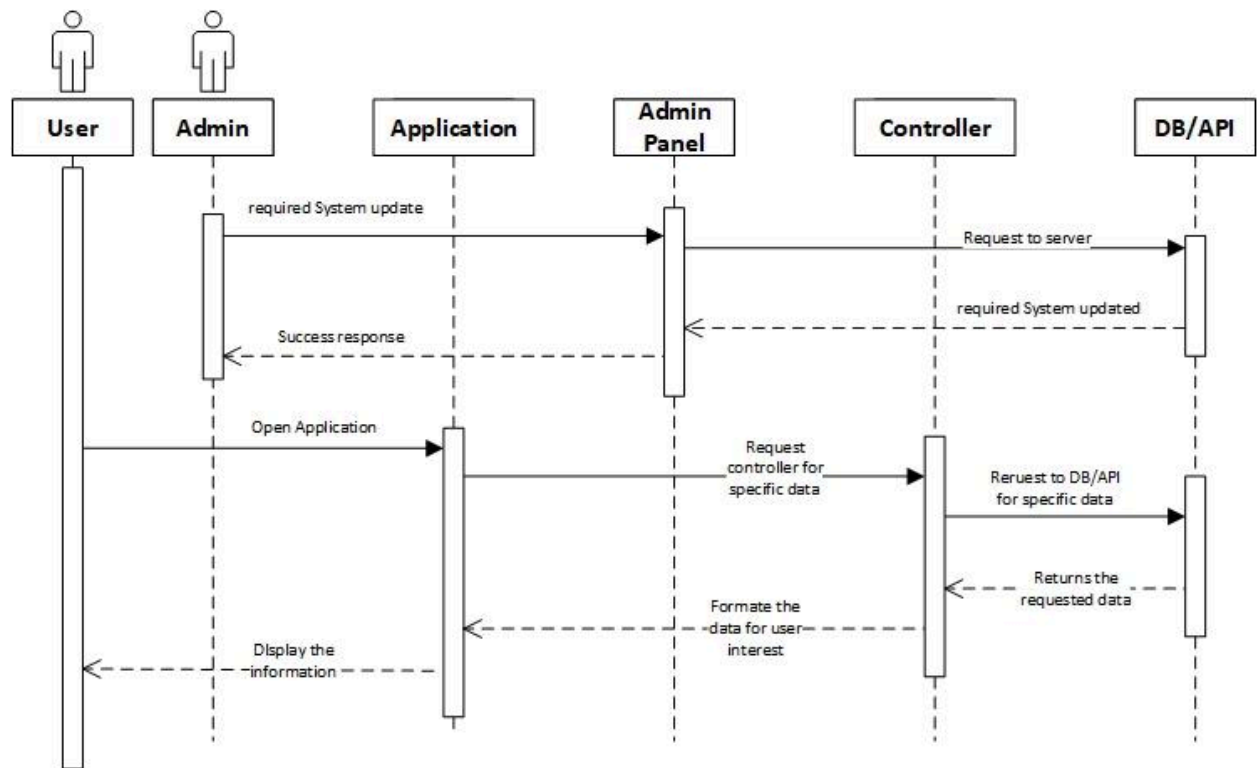


Figure-3: Sequence Diagram

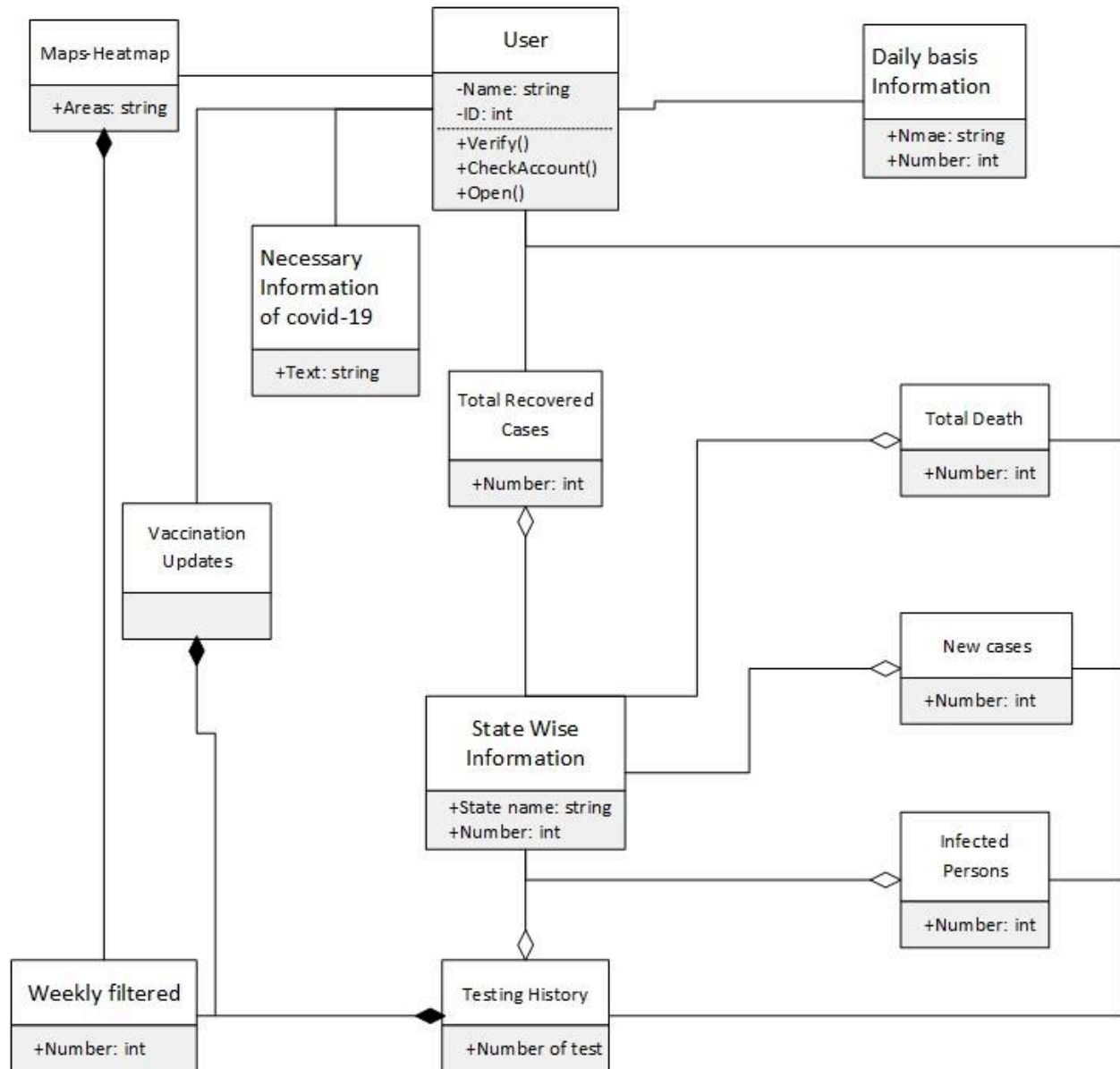


Figure-3: Class Diagram

HMI design for SCS :

Human Machine Interface design for safety critical systems is described according to the paper named as "Human-Machine Interface: Design Principles of Visual Information in Human-Machine Interface Design" [1].

Requirements:

- The visualizable human-machine interface should be used to provide the consumers specific and identifiable objects, helping the consumers induce abstract concepts or principles
- Taking advantage of the visualizable interface to demonstrate the hidden contents, and making full use of distinct and intelligible objects such as characters, graphics, animation, colors, etc. to demonstrate principles, formulas and abstract concepts
- Using the visualizable human-machine interface design to stimulate the consumers' thoughts and imagination, so as to enhance their participation and stimulate their desire to study and create.

Properties:

- Recognition
- Interaction
- Maneuverability
- Affinity

Principles:

- Reasonable adoption of visual elements
- Avoiding visual disorder
- Main elements and backgrounds distinguishable

Background Color	Main color
Black	Yellow
White	Black
Yellow	Black
Black	White
Purple	Yellow
Purple	White
Blue	White
Green	White
Yellow	Green
Yellow	Blue

Table : Easy to distinguish collocation of color Space-coordination principle

Background Color	Main color
Yellow	White
White	Yellow
Purple	Black
Black	Blue
Gray	Green
Red	Purple
Green	Red
Red	Blue
Red	Yellow
Black	Purple

Table : Difficult to distinguish collocation of color Space-coordination principle

Color	Psychological reaction
Red	Excited, enthusiastic, brilliant, dangerous ...
Orange	Moderate, pleasant, bright, irritable ...
Yellow	Warmth, harvest, hope, cheerfulness, commonness ...
Green	Vitality, growth, hope, fresh ...
Blue	Equability, calm, far-reaching, cold, mind ...
Purple	Noble, luxurious, elegant, mysterious ...
Black	Dignified, solid, heavy, terrorist ...
White	Immaculacy, holiness, lustration, pure, crisp, true ...
Grey	Simple, light, soft ...

Table: Color and their psychological reaction

Safety-critical HMI implications of our use cases

- User is getting the right and updated data(COVID-19) from the system
- User is getting the all COVID-19 cases graphs showing accurate results respective to the states
- No user's personal information is collected
- Error free controller at the time of uploading the API
- User is getting the rating and review of medical and medicine organizations which are shared by mass public
- The system or the web interface is always accessible by the users from all over the world
- Reasonable adoption of visual elements
- Avoiding visual disorder
- Main elements and backgrounds distinguishable
- Reasonable adoption of colors
- Reasonable layout and easy access of data

Our HMI design rules

- Don't make the user think
- Keep the user in control
- Design for interaction
- Break up dense content into chunks
- Group related elements
- Strong visual hierarchy
- Focus on one action at a time
- Imagery should have a clear purpose
- Match the user's mental model
- Predictable

Our Application's Color Code:

Web Application

Navbar:



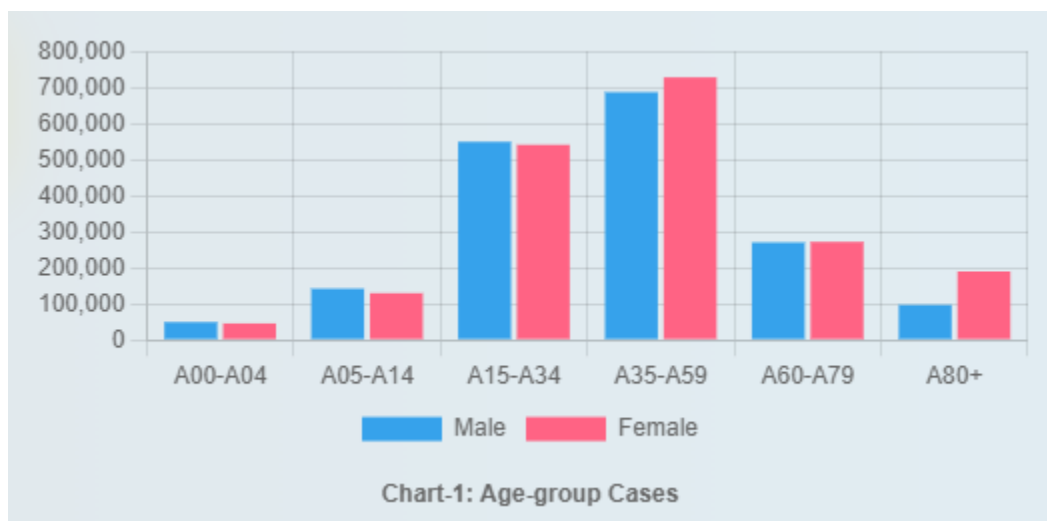
Total cases & deaths:

Total Cases	Total Deaths	Total Recovered	Last Week Cases
3727333	90819	3622612	4468

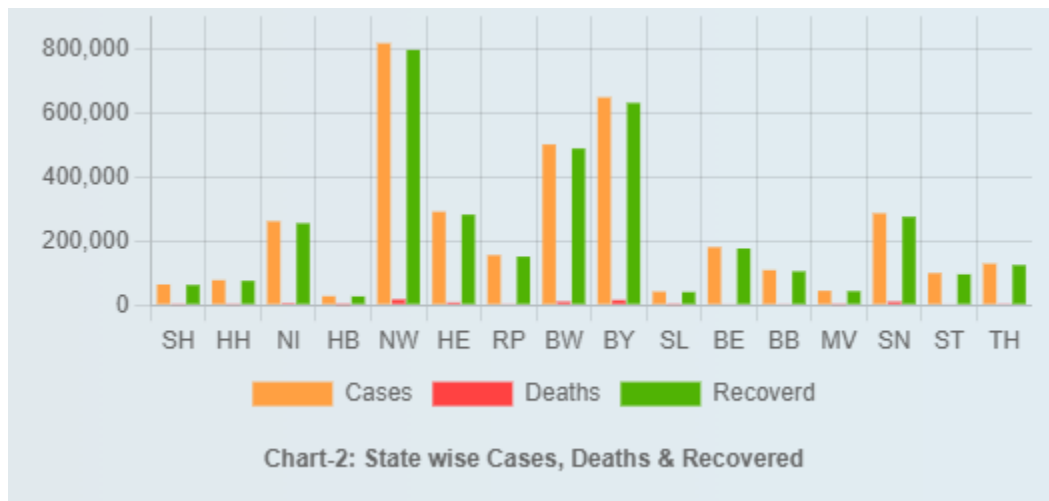
Last 7 days information:

Last 7 days Information		
Date	Cases	Recovered
2021-06-23	903	28
2021-06-24	807	53
2021-06-25	739	79
2021-06-26	477	34
2021-06-27	197	24
2021-06-28	296	32

Age-grouped cases:



State wise cases, deaths & recovered:



Mobile Application

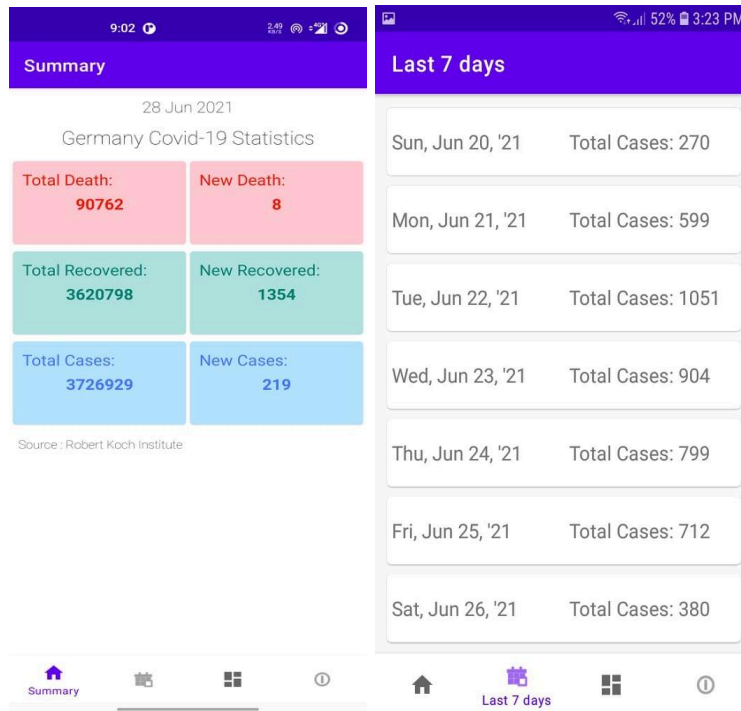
Starting page:



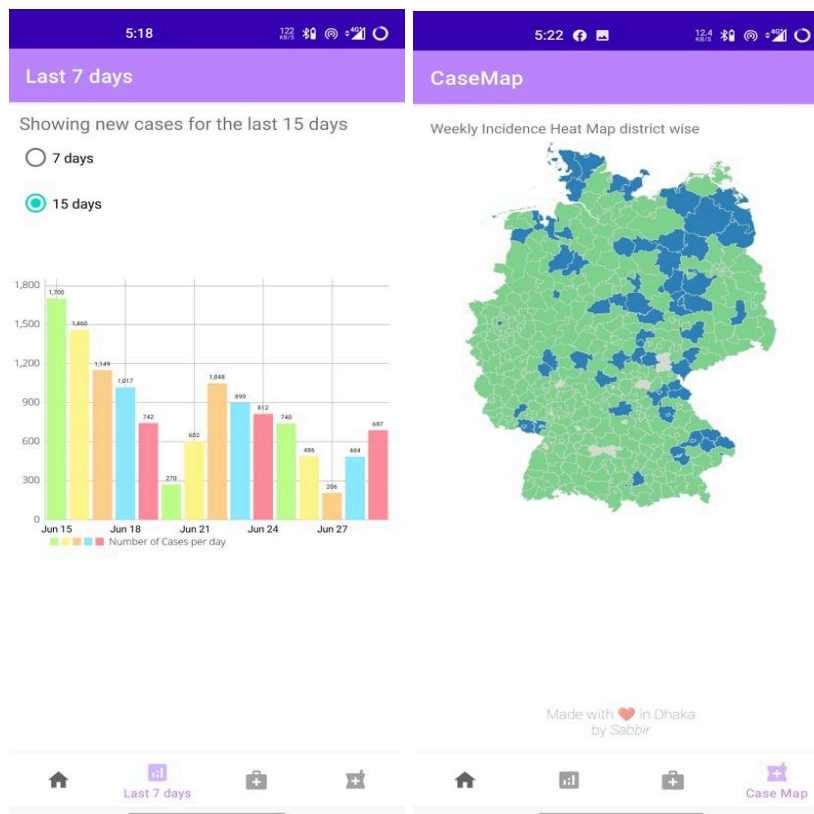
Covid-19 Information
Germany



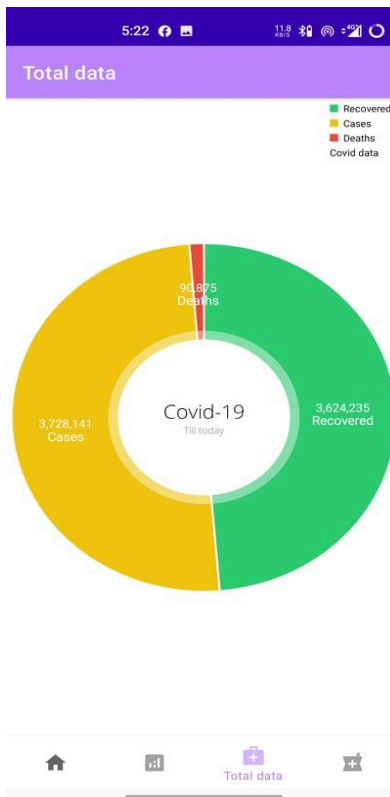
Summary & 7-days data:



Bar Chart & heat-map:



Total data in pie chart:



2. Human Error Models:

In the lectures the Rassmussen Model is explained. Please read and summarize the following paper: Human error taxonomies applied to driving:

A generic driver error taxonomy and its implications for intelligent transport systems by NA Stanton, PM Salmon - Safety Science, 2009 - Elsevier.

The part of human blunder in mishaps in most wellbeing basic frameworks is notable. For instance, inside common avionics human blunder has been recognized as a causal factor in around 75% of all mishaps, what's more, is presently seen as the essential danger to flight security (Civil Aviation Authority, 1998). Examination concerning the build has prompted the improvement of blunder centered mishap examination and investigation strategies, like the human components examination and arrangement framework. This was a test of contemporary thinking, and features the significance of plan in human mistake decrease. He got keen on why pilots regularly withdrew the landing gear rather than the setting down folds subsequent to setting down the airplane.

The utilization of formal human blunder order plans is boundless all through most complex wellbeing basic frameworks. Human mistake characterization plans are utilized both supportive of effectively, to

expect blunders that may happen, and reflectively, to order and break down blunders that have happened during mishaps and occurrences. The forecast of human mistake is accomplished using formal human blunder ID (HEI) methods, for example, the methodical human mistake decrease approach (SHERPA; Embrey, 1986), which utilizes a scientific categorization of outside blunder modes (EEMs) to recognize blunders that might actually happen during task execution.

Likely the most generally detailed driver mistake study was led by Reason et al. (1990) who looked to make a differentiation between driver mistakes and infringement. Blunders can be characterized as events where the driver's proposed execution was acceptable, however it as a matter of fact missed the mark, (for example, meaning to drive inside as far as possible, however inadvertently squeezing the gas pedal excessively far (a slip), failing to remember as far as possible (a pass), or imagining that as far as possible is 70 mph when it is really 60 mph (a misstep). Interestingly, purposeful infringement might be characterized as events where the driver's aims were to play out the activity, for example, intentionally surpassing as far as possible. Reason et al. (1990) fostered the driver conduct survey (DBQ), a 50-thing poll involving five classes of unusual driver conduct: slips; slips; botches; accidental infringement; and purposeful infringement. The investigation of blunders and infringement utilized the DBQ, which inspected 520 drivers in nine age groups, from under 20 years to over 56 years. Drivers were approached to report the recurrence with which they submitted various kinds of mistakes and infringement while driving. The examination was embraced in light of a call for better grouping frameworks for mishap agents. Table 3 presents model driving mistakes identified with Reason's (1990) blunder and infringement scientific categorization.

Almost certainly, the kinds of driver mistakes portrayed in this paper will keep on being seen in street auto collisions and occurrences. A portion of these mistakes may affect new vehicle advances essentially for the driver of the host vehicle. To educate the plan of future in-vehicle frameworks (likewise vehicle frameworks) and examination and investigation of human blunder inside the street transport area, scientific categorizations of driver mistakes and their causal components are required. A driver mistake scientific categorization might actually be utilized to distinguish, deduce, driver blunders and furthermore to order the mistakes engaged with street transport mishaps and occurrences. A causal variables scientific classification could be utilized to advise the turn of events of mistake the board systems and blunder countermeasures, and likewise to group the causal variables associated with driver blunder related episodes.

There is incredible potential for driving advances to be utilized to annihilate driver mistakes or to moderate their results. For model, canny vehicle frameworks (ITS, for example, course route frameworks, versatile voyage control frameworks, and shrewd speed transformation frameworks might all actually be utilized to either decrease blunder event by keeping a driver from playing out a mistaken activity, or alleviate the results related with mistakes by expanding the resistance of the vehicle to driver mistakes. For each of the mistakes introduced in Table 11, a potential innovative arrangement has been allocated in Table 13. The suitability of driver conduct with cutting edge innovations will, to some degree, rely on the plan of the interface between the driver and the conduct of the framework. The driver is needed to expect and anticipate the conduct of the framework. This will rely on that person fostering an exact mental

portrayal and monitoring what the framework is doing anytime on schedule. Woods et al. (1994) contend that aversion of mode mistake furthermore, advancement of situational mindfulness go inseparably. They recommend that plan of mechanized frameworks ought to:

1. kill pointless modes;
2. give obvious signs of mode status;
3. give input about mode changes; and
4. be open minded toward mode mistake if conceivable.

The plan of an unambiguous interface that imparts the status of the framework in an immediate way tends to the center two focuses. Likewise with pilot blunder, the test for the architects will be to present advancements that really lessen driver blunder, without making the opportunities for new sorts of blunder.

3(a) problems of n-version programming:

N-version programming has been proposed as a method of incorporating fault tolerance into software Critical software applications, such as mission-critical and, in particular, safety-critical applications, have the constraint that they cannot fail. The nature of software makes this a very difficult objective to achieve in most cases. In hardware design, one approach is to introduce redundant hardware components into highly reliable hardware-based systems. These redundancies of the components are taken advantage of by asking each component to do a computation. The individual results are then compared and the ultimate outcome is determined by taking a vote based on the results, with the majority winning. The premise is that the failure of individual hardware components is independent of the other components in the system. That is, the failure of one component in no way influences the failure of another. These ideas have been introduced into software design in the form of n-version programs. The idea is that for a particular software component, n versions of the component will be built independently and integrated into the system along with a voting mechanism. The voting mechanism takes the output of each redundant component, compares them, and determines what the result should be based on some algorithmic voting process. Differences in the results reported by the redundant components result from some failure in one or more components. The problem with an n-version approach to developing software components is that their failures may not be independent. If this is true, then the results from an n-version software computation would give a false sense of security.

3(b):Use of n-version programming and recovery blocks:

N-version programming and recovery blocks has been used for software in

Switching trains

Performing flight control computations on modern airliners

Electronic voting (the SAVE System)

The detection of zero-day exploits

4. Please summarize the main points of the round table discussion by Jeffrey Voas! The discussion is getting right to the main points!

Fault Tolerance:

This is a "virtual" roundtable conversation between five regarded specialists from the open minded registering field: Joanne Bechta Dugan (UVA), Les Hatton (Oakwood Computing), Karama Kanoun and Jean-Claude Laprie (LAAS-CNRS), and Mladen Vouk (NC State College). The justification including this piece is to give a little instructional exercise to users who are new to the field of programming adaptation to internal failure or who may have had more openness to equipment adaptation to internal failure. Eleven questions were presented to every individual from the board over email, and here we present their reaction.

A primary method of bringing adaptation to non-critical failure into a framework is to give a technique to progressively decide in the event that the framework is acting as it ought to—that is, you present a self-checking or "prophet" capacity. On the off chance that the technique recognizes sudden what's more, undesirable conduct, a shortcoming lenient framework should give the way to recuperate or proceed with activity (ideally, from the client's point of view, in a consistent way). Equipment adaptation to non-critical failure, generally, manages arbitrary disappointments that outcome from equipment abandons happening during framework activity. Programming adaptation to internal failure fights with irregular (and some of the time not-so random) summons of programming ways (or then again way and state or climate mixes) that generally actuate programming plan and execution abandons.

These deformities would then be able to prompt framework disappointments. Runtime disappointment location is regularly refined through by the same token an acknowledgment test or correlation of results from a mix of "various" yet practically identical framework substitutes, parts, forms, or on the other hand

variations. In any case, different strategies—going from numerical consistency checking to blunder coding to information variety—are additionally valuable. There are numerous choices for successful framework recuperation after an issue has been distinguished. They range from complete restoration (for instance, halting with a full information and programming reload and afterward restarting) to dynamic forward mistake amendment to halfway state rollback and restart. Hazard investigation will yield the issue event to cost-to-profit results, which we would then be able to use to settle on fitting choices. : Usually, the prerequisite is less for adaptation to non-critical failure (without anyone else) all things considered for high accessibility, unwavering quality, and security. Henceforth, IEEE, FAA, FCC, DOE, and different norms and guidelines fitting for solid PC based frameworks apply. We can accomplish high accessibility, dependability, and security in an unexpected way. They include a legitimate solid and safe plan, appropriate shields, and legitimate execution. Adaptation to non-critical failure is only one of the procedures that guarantees that a framework's nature of administration (from a more extensive perspective) addresses client issues (like high wellbeing). You realize you've accomplished adaptation to non-critical failure through a blend of good prerequisites particulars, sensible quantitative accessibility, unwavering quality and wellbeing boundaries, intensive plan investigation, formal strategies, and genuine testing. These assumptions are probably going to begin stretching out to additional everyday things like individualized computing and systems administration applications, vehicle computerization, (for example, GIS finders), what's more, maybe, space, the travel industry. Nearly none of these would gauge up today, and many will require new fault tolerance moves toward that are a long way from standard course book arrangements.

References:

1. Gong, C. (2009). Human-Machine Interface: Design Principles of Visual Information in Human-Machine Interface Design. 2009 International Conference on *Intelligent Human-Machine Systems and Cybernetics*. doi:10.1109/ihmsc.2009.189