**GUJARAT TECHNOLOGICAL UNIVERSITY**
**C.K.PITHAWALA COLLEGE OF ENGINEERING & TECHNOLOGY, SURAT**
**COMPUTER ENGINEERING DEPARTMENT**

| **PRACTICAL** | |
|---|---|
| **Name of Student** | Shah Dhruvil N. |
| **Enrollment No.** | 170090107050 |
| **Subject Name** | Software Engineering |
| **Subject Code** | 2160701 |
| **Student Class** | 3rd Year (6th Semester) |
| **Academic Year** | 2019-20 |

**C.K.PITHAWALA COLLEGE OF ENGINEERING & TECHNOLOGY, SURAT**
**COMPUTER ENGINEERING DEPARTMENT**

# CO and PSO

## Course Outcome (CO)

| Course Outcome | CO Statement |
|---|---|
| CO1 | Understand the basic concepts and various software process models (conventional and agile) with industrial approach. |
| CO2 | Acquire knowledge for project management activities such as planning, scheduling, tracking and risk management. |
| CO3 | Develop software requirements, design, architecture systematically using standard tools and methodologies. |
| CO4 | Gain knowledge of coding standards and various testing techniques. |
| CO5 | Study different quality standards and software review methods. |
| CO6 | Familiar with advanced trends in Software Engineering. |

## Program Specific Outcome (PSO)

| Program Outcome | PSO Statement |
|---|---|
| PREPARATION | To prepare students to gain adequate knowledge in the field of computer engineering for successful career in industry, research and higher studies |
| CORE COMPETENCE | Instilling core competence and capability in graduates to solve real world engineering and societal problems |
| BREADTH | To provide opportunity to the student to interpret, analyze and design computer programs for interdisciplinary projects |
| PROFESSIONALISM | To develop the qualities of team work, communication skills, entrepreneurship and work ethics following the codes of professional practice |
| LEARNING ENVIRONMENT | To promote lifelong self-learning capability among graduates to cope up with technological changes |

# Gujarat Technological University
# C. K. Pithawalla College of Engineering and Technology, Surat
# Academic Year : 2019-20
# B.E. - III Semester –VI (Computer Engineering)
# Subject Name: Software Engineering (2160701)

## Practical List

## NOTE :

- Select Any Software System to be developed in any technology and perform following activities.
- Refer Virtual Lab : http://vlabs.iitkgp.ernet.in/se/

| Sr No | Problem Statement | Cos |
|---|---|---|
| 1. | **Software Development Life Cycle**<br>Design Software Development Life Cycle (SDLC) model and analyse various activities conducted as a part of various phases. For each SDLC phase, identify the objectives and summaries outcomes. | **CO1** |
| 2. | **Requirement Analysis and Specification** | **CO3** |
| | Prepare Software Requirement Specification (SRS) document. | |
| 3. | **Design Analysis** | **CO3** |
| | 3.1 Draw E-R Diagram for your system. | |
| | 3.2 Design UML Diagrams<br>3.2.1 Use Case Diagrams<br>3.2.2 Class Diagram and Sequence Diagram<br>3.2.3 Activity Diagram and State Diagram | |
| | 3.3 Design DFD level -0 and level-1 for your system. | |
| | 3.4 Prepare User interface design (Front end) and also mention characteristics of it. | |
| 4. | **Software Project Management** | **CO2** |
| | Examine Project Management Tool and prepare a report. | |
| 5. | **Coding and Testing** | **CO4** |
| | 5.1 Implementation<br>5.2 Examine various code analysis tools and prepare a comparative analysis.<br>5.3 Examine various Testing tools (Win runner, Load runner) and prepare a comparative analysis. | |

**Faculties Name:** **DIC**

Prof. Vishruti Desai
Prof. Unnati Shah

Prof. Neelam Surti

# 1. Software Development Life Cycle

Design Software Development Life Cycle (SDLC) model and analyse various activities conducted as a part of various phases. For each SDLC phase, identify the objectives and summaries outcomes

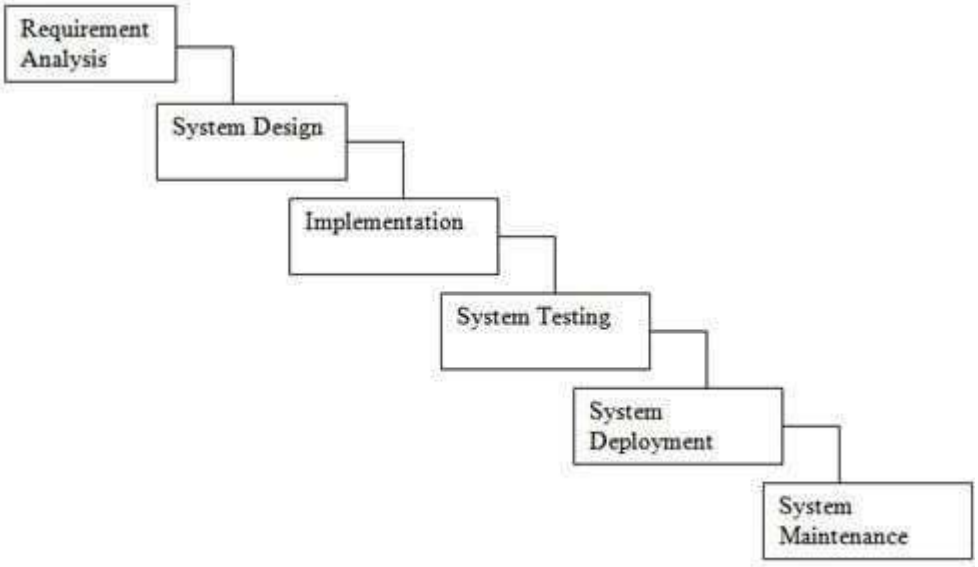
- **List of Models**
  - Waterfall Model
  - Prototyping Model
  - Incremental Model
  - Spiral Model
  - RAD (**R**apid **A**pplication **D**evelopment)

# • Waterfall Model

## ➢ What is Waterfall Model?

- In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.
- The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

## ➢ Diagram




Waterfall Model - © www.SoftwareTestingHelp.com

## ➢ Phases

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next

phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## ➢ Advantages

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

## ➢ Disadvantages

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

## ➢ Where we can use?

- **Waterfall used when,**
- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.

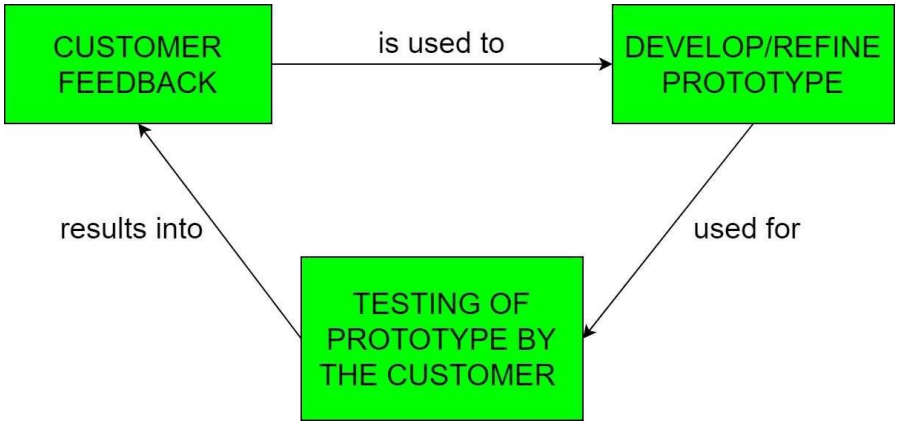
- Ample resources with required expertise are available to support the product.
- The project is short.

## • Prototyping Model

### ➢ What is Prototyping Model?

- This model is used when the customers do not know the exact project requirements beforehand.
- In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.

### ➢ Diagram




### ➢ Steps

- **Basic Requirement Identification** - This step involves understanding the very basics product requirements especially in terms of user interface.
- **Developing the initial Prototype** - The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed.
- **Review of the Prototype** - The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an organized manner and used for further enhancements in the product under development.
- **Revise and Enhance the Prototype** - The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until the customer expectations are met.

## ➢ Advantages

- The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
- New requirements can be easily accommodated as there is scope for refinement.
- Missing functionalities can be easily figured out.
- Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.
- The developed prototype can be reused by the developer for more complicated projects in the future.
- Flexibility in design.

## ➢ Disadvantages

- Costly w.r.t time as well as money.
- There may be too much variation in requirements each time the prototype is evaluated by the customer.
- Poor Documentation due to continuously changing customer requirements.
- It is very difficult for the developers to accommodate all the changes demanded by the customer.
- There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted by the customer.
- After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.
- Developers in a hurry to build prototypes may end up with sub-optimal solutions.
- The customer might lose interest in the product if he/she is not satisfied with the initial prototype.

## ➢ Where we can use?

- The Prototyping Model should be used when the requirements of the product are not clearly understood or are unstable.
- It can also be used if requirements are changing quickly. This model can be successfully used for developing user interfaces, high technology software- intensive systems, and systems with complex algorithms and interfaces.
- It is also a very good choice to demonstrate the technical feasibility of the product.

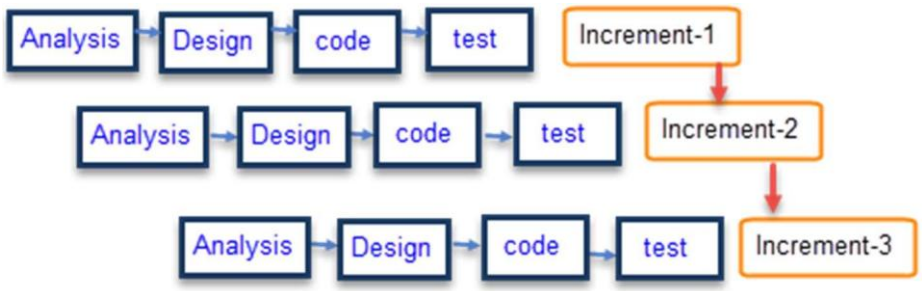
# • Incremental Model

## ➢ What is Incremental Model?

- Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle.
- Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.

## ➢ Diagram

- Each iteration passes through the **requirements, design, coding and testing phases**. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.




Incremental Model

## ➢ Phases

- **Requirement Analysis -** Requirement and specification of the software are collected.
- **Design -** Some high-end functions are designed during this stage.
- **Code -** Coding of software is done during this stage
- **Test -** Once the system is deployed, it goes through the testing phase

## ➢ Advantages

- The software will be generated quickly during the software life cycle
- It is flexible and less expensive to change requirements and scope
- Throughout the development stages changes can be done
- This model is less costly compared to others
- A customer can respond to each building
- Errors are easy to be identified

- **Disadvantages**
  - It requires a good planning designing
  - Problems might cause due to system architecture as such not all requirements collected up front for the entire software lifecycle
  - Each iteration phase is rigid and does not overlap each other
  - Rectifying a problem in one unit requires correction in all the units and consumes a lot of time

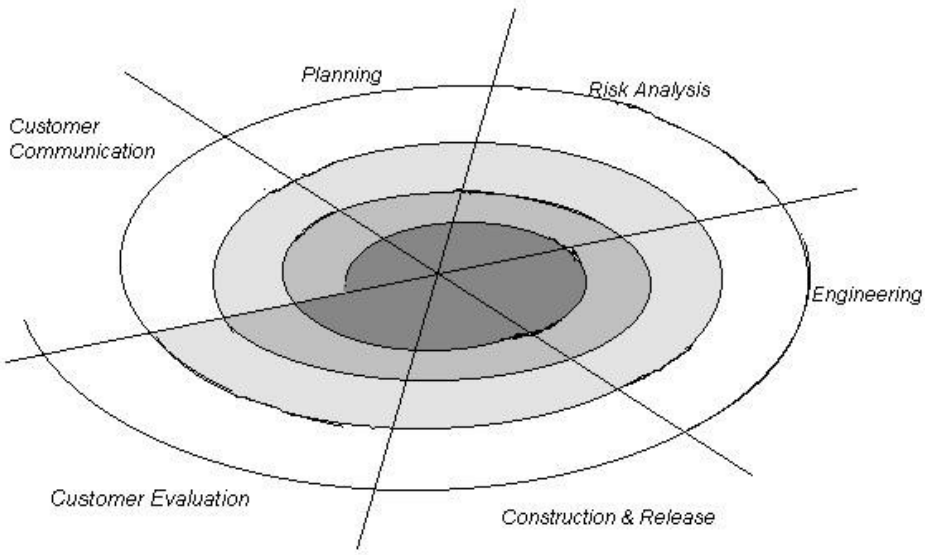
- **Where we can use?**
  - Requirements of the system are clearly understood
  - When demand for an early release of a product arises
  - When software engineering team are not very well skilled or trained
  - When high-risk features and goals are involved
  - Such methodology is more in use for web application and product based companies.

# • Spiral Model

## ➢ What is Spiral Model?

- It provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops.
- Each loop of the spiral is called a Phase of the software development process**.** The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

## ➢ Diagram




## ➢ Phases

- **Planning -** It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer.
- **Risk Analysis -** Identification of potential risk is done while risk mitigation strategy is planned and finalized.
- **Engineering -** It includes testing, coding and deploying software at the customer site.
- **Evaluation -** Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun.

## ➢ Advantages

- Additional functionality or changes can be done at a later stage.
- Cost estimation becomes easy as the prototype building is done in small fragments.
- Continuous or repeated development helps in risk management.
- Development is fast and features are added in a systematic way.

- There is always a space for customer feedback.

## ➢ Disadvantages

- Risk of not meeting the schedule or budget.
- It works best for large projects only also demands risk assessment expertise.
- For its smooth operation spiral model protocol needs to be followed strictly.
- Documentation is more as it has intermediate phases.
- It is not advisable for smaller project it might cost them a lot.

## ➢ Where we can use?

- When project is large
- When releases are required to be frequent
- When creation of a prototype is applicable
- When risk and costs evaluation is important
- For medium to high-risk projects
- When requirements are unclear and complex
- When changes may require at any time
- When long term project commitment is not feasible due to changes in economic priorities

# • RAD

# • (Rapid Application Development)

## ➢ What is RAD?

- RAD model is Rapid Application Development model. It is a type of incremental model.
- In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype.
- This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

## ➢ Diagram

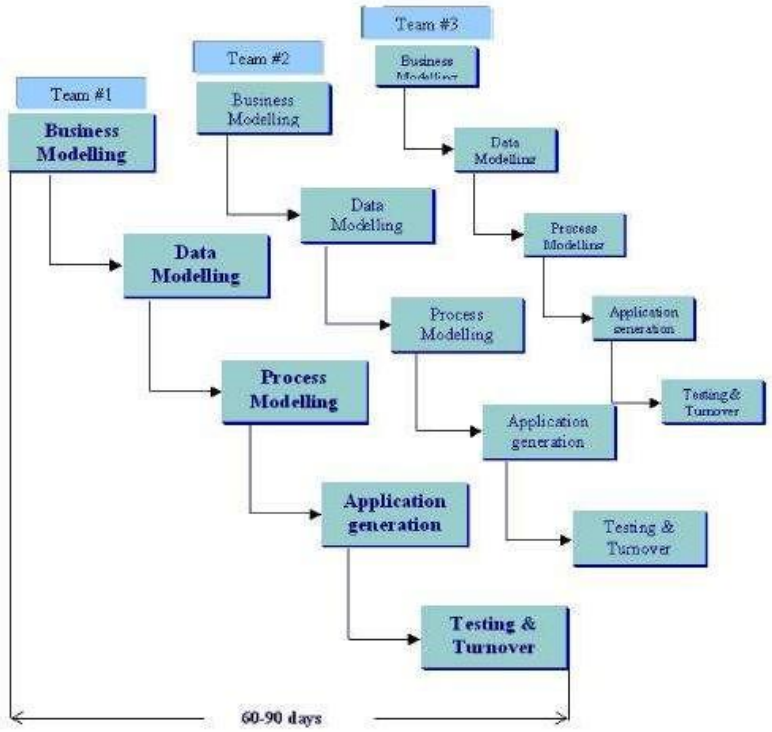



Figure 1.5 – RAD Model

## ➢ Phases

- **Business modelling-** The information flow is identified between various business functions.
- **Data modelling-** Information gathered from business modeling is used to define data objects that are needed for the business.
- **Process modelling-** Data objects defined in data modeling are converted to

achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects

- **Application generation-** Automated tools are used to convert process models into code and the actual system.
- **Testing and turnover -** Test new components and all the interfaces.

## ➢ Advantages

- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

## ➢ Disadvantages

- Depends on strong team and individual performances for identifying business requirements.
- Only system that can be modularized can be built using RAD
- Requires highly skilled developers/designers.
- High dependency on modelling skills
- Inapplicable to cheaper projects as cost of modelling and automated code generation is very high.

## ➢ Where we can use?

- RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
- It should be used if there's high availability of designers for modelling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
- RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

# 2. Requirement Analysis and Specification

## SOFTWARE REQUIREMENTS SPECIFICATION

## Project Title : Digital Id Proof System

**Subject : Software Engineering**

**Subject Code : 2160701**


| Sr No | Enrollment Number | Name |
|---|---|---|
| 1 | 170090107050 | Shah Dhruvil N |
| 2 | 170090107051 | Shah Keneel C |


Faculty Name :
Prof. Unnati Shah
Prof. Vishruti Desai

Signature : **Date**

# Software Requirements Specification

---

## eDocs
## (Digital ID Proof Management Sysrem)

Prepared By:

Dhruvil Shah (170090107050)
Keneel Shah (170090107051)

# Table of Contents

# 1. Introduction

## 1.1Purpose

The purpose of this document is to provide detailed information about the requirements for the "Digital ID Proof Management System" (eDocs) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## 1.2Scope

eDocs is a web portal which helps people to keep and manage all the Government issued document at one place. The web portal will be free to use.

The Documents will be provided by the government. During the time of registration, the user will have to enter his/her unique ID provided by UIDAI to create an account on the website.

On the website the user will be able to add his/her Government Approved ID proofs using the number of the ID proof. Once added to the profile the user will be able to access it anytime and from anywhere.

## 1.3Definitions, Acronyms, and Abbreviations

- DIPS – Digital Id Proof System
- DIPS is also referred as eDocs or edocs.
- DFD – Data Flow Diagram
- ERD – Entity Relationship Diagram
- SRS – Software Requirements Specification
- User – The people who will avail this facility.
- Govt. – Government

## 1.4 References

- Internet resources
  - DIGILOCKER – https://digilocker.gov.in/resource/technical-specifications.html
- Books
  - Software engineering – Shalini puri
  - Software engineering – Pressman

## 1.5 Overview

eDocs is a service to provide a secure dedicated personal electronic space for storing the Govt. Issued documents of users such as PAN cards, Voter ID cards and Aadhar card. The service is intended to minimize the use of physical documents and to provide authenticity of the documents. Whenever the users want to share their documents with others, they can share them within a fraction of seconds. It maintains all data of User's

- Personal Account information.
- Aadhar card.
- Voter-Id card.
- Pan card.

# 2. General Description

- This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution.
- It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements workshop.
- It further lists and briefly describes the major features and a brief description of each of the proposed system.

## 2.1 Product Perspective

✓ Before the automation, the system suffered from the following **DRAWBACKS**:

- The existing system is highly manual involving a lot of paper work and calculation therefore may be erroneous. This has lead to inconsistency and inaccuracy in the maintenance of data.
- The data, which is stored on the paper only, may be lost, stolen or destroyed due to natural calamity like fire and water.
- The existing system is sluggish and consumes a lot of time causing inconvenience to users.
- Due to manual nature, it is difficult to update, delete, add or view the data.
- Since the number of citizens have drastically increased therefore maintaining and retrieving detailed record of passenger is extremely difficult.

✓ Hence the **Benefits** of Digital Id Proof System is proposed with the following

- Prevents Eliminate need for the residents to maintain hard copy of government issued documents.
- Eliminate need for the residents to produce (in hard format) government issued documents, while applying for services.
- Provide secure and consented access of government issued documents to user agencies.
- Reduce administrative burden, service fulfillment time, and costs by enabling paperless transactions.
- Ensure all the documents issued to the residents are available to him/her anywhere anytime, in a standard format which can be shared with any other department.
- Provide an open, interoperable, multi-provider architecture to ensure departments and states have flexibility to use best document repository for their purposes.
- Provide an architecture that can support well structured future documents as well as a mechanism to digitize older documents that may not have machine readable formats.
- Provide a default portal and mobile application for residents to view their documents in a consolidated way.

## 2.2 Product Functions

- The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

1. Users-
    a. In this module users has to register with Aadhar id or mobile number if he/she verified than, then they can login and retrieve the his/her's private documents such as Voter id,

Aadhaar card, Pan card and Driving license etc., to their account according to the request he/she make by entering respective unique Id of document for retrieval.

b. Users can share their documents with anyone on any platform after downloading it.
c. Users can view and download the documents issued by the admin through Government database.
d. If any user loses their physical documents then he/she can download from DIPS software and use it as temporary solution.

2. Admin-
   a. Admin have report of the documents retrieved by any user.
   b. Admin handles documents such as Aadhar card, Pan Card, Voter-Id card, Driving license etc., which are requested by the user using government database.
   c. Admin can view all registered users

The scope of this project encompasses: -

- **Search**: System design is to create a technical solution that satisfies the functional requirements for the system. At this point in the project life cycle there should be a Functional Specification, written primarily in business terminology, containing a complete description of the operational needs of the various organizational entities that will use the new system. The challenge is to translate all of this information into Technical Specifications that accurately describe the design of the system, and that can be used as input to System Construction.

- **Selection**: This function allows a particular document to be selected from the displayed list. All the details of the documents are shown :-

1. Select option for Aadhar card to retrieve Aadhar document of the user
2. Select option for Pan Card to retrieve Pan document of the user
3. Select option for Voter-Id Card to retrieve Voter-Id document of the user
4. Select option for Driving license to retrieve Driving license document of the user
5. View Option for viewing retrieved document for user.

- **Users Information**: It asks for the details of all the users including Full name, Mobile number , Aadhar Card's Unique Id, Gender, Password for users verification through Aadhar card unique Id.

- **Retrieval**: DIPS software request user to enter the Unique Id card details of documents for which he/she is searching for.

1. Aadhar Card's Unique Id Number
2. Pan Card Unique Id Number
3. Voted-id Card Unique Id Number
4. Driving license Card Unique Id Number

## 2.3 User Characteristics

- At least user of the system should be comfortable with English language.
- User should be comfortable using general purpose applications on the computer system.

## 2.4 General Constraints

- Application must follow the proper Database access.
- Application should able to save and display all requested document by user on his/her

dashboard.

## 2.5 Assumptions and Dependencies

- User will be having a valid Aadhar Card Unique Id Number/ Mobile Number and password to access the software.
- The software needs user to have knowledge of Digital Id Proof System Before Using Which information is accessible to user through buttons provided on home page like FAQ's and Quick info.
- Software is dependent on access to internet.

## 3. Specific Requirements

- This section contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system and all functions performed by the system in response to an input or in support of an output. The following principles apply:

*(1)* Specific requirements should be stated with all the characteristics of a good SRS
- correct
- unambiguous
- complete
- consistent
- ranked for importance and/or stability
- verifiable
- modifiable
- traceable

*(2)* Specific requirements should be cross-referenced to earlier documents
*(3)* Types of information used by various functions
*(4)* Frequency of use
*(5)* Accessing capabilities
*(6)* Data entities and their relationships
*(7)* Integrity constraints
*(8)* Data retention requirements
*(10)* If the customer provided you with data models, those can be presented here. ER diagrams (or static class diagrams) can be useful here to show complex data relationships. Remember a diagram is worth a thousand words of confusing text.
*(11)* that relate
*(12)* All requirements should be uniquely identifiable
*(13)* Careful attention should be given to organizing the requirements to maximize Readability (Several alternative organizations are given at end of document)

### 3.1 Functional Requirements

1. Admin should be able to
    a. login to the system through the first page of the application
    b. Admin have access of government database so admin can manage the document when user request for any particular type of document.
    c. Admin have access to database containing information of registered users on Digital Id Proof System.
    d. View documents which are retrieved by the users.
    e. Send the documents like Pan card, Aadhar card, Voter-Id card, driving license etc., that are requested by the users.

2. Users/Citizens should be able to
    a. Register and login to the system through the first page of the application

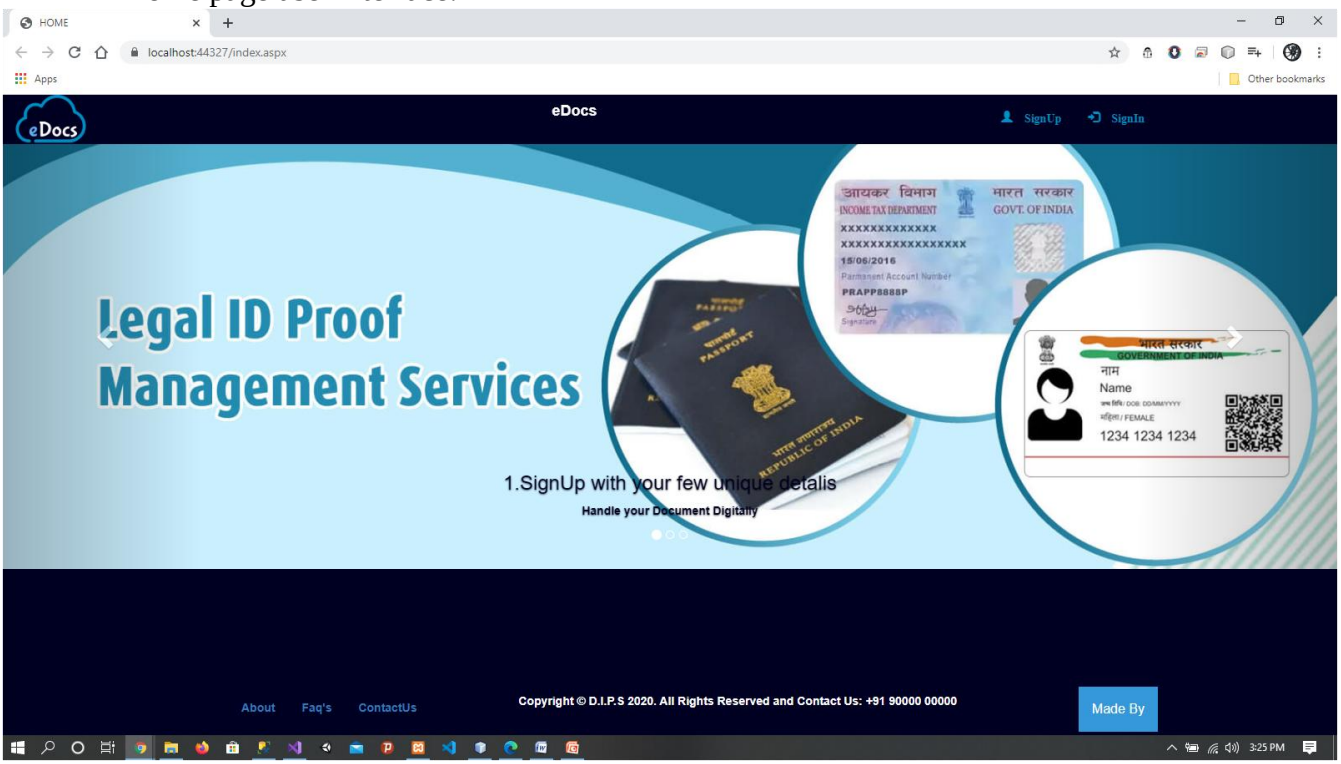
b. Retrieve their private documents and store them on their dashboard so problem of carrying physical documents gets solved.
c. Share documents with admin as well as with to any platform he/she needs.
d. View and download documents issued by the admin trough government access database.
e. Can request for other documents like Pan card, Aadhar card, Voter-Id card, driving license etc., by filling Unique Id respected to which document he/she wants to retrieve.
f. Can notify about lost document.

**Non-Functional Requirements**

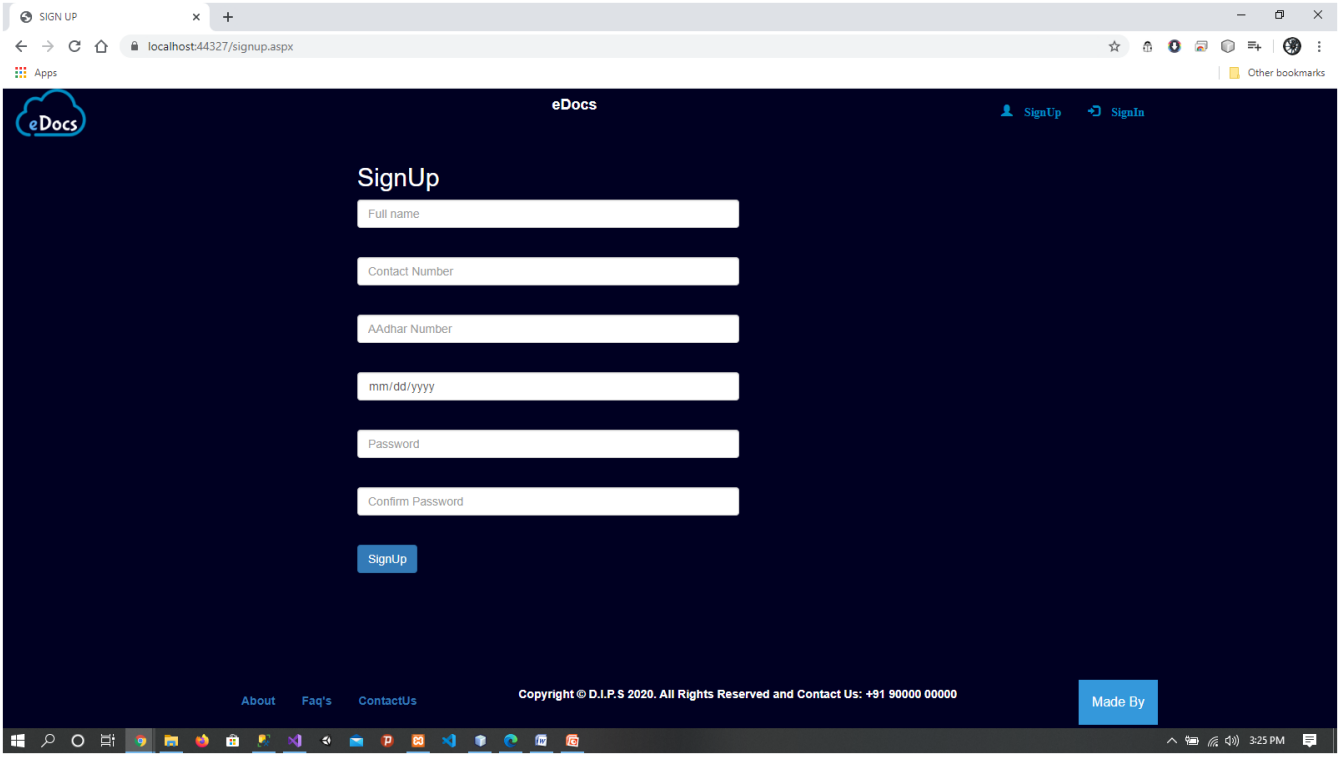
1. The system shall be user friendly and consistent
2. The system shall provide attractive graphical interface for the user
3. The system shall allow developer access to installed environment
4. The system shall target customer base

### 3.2 External Interface Design

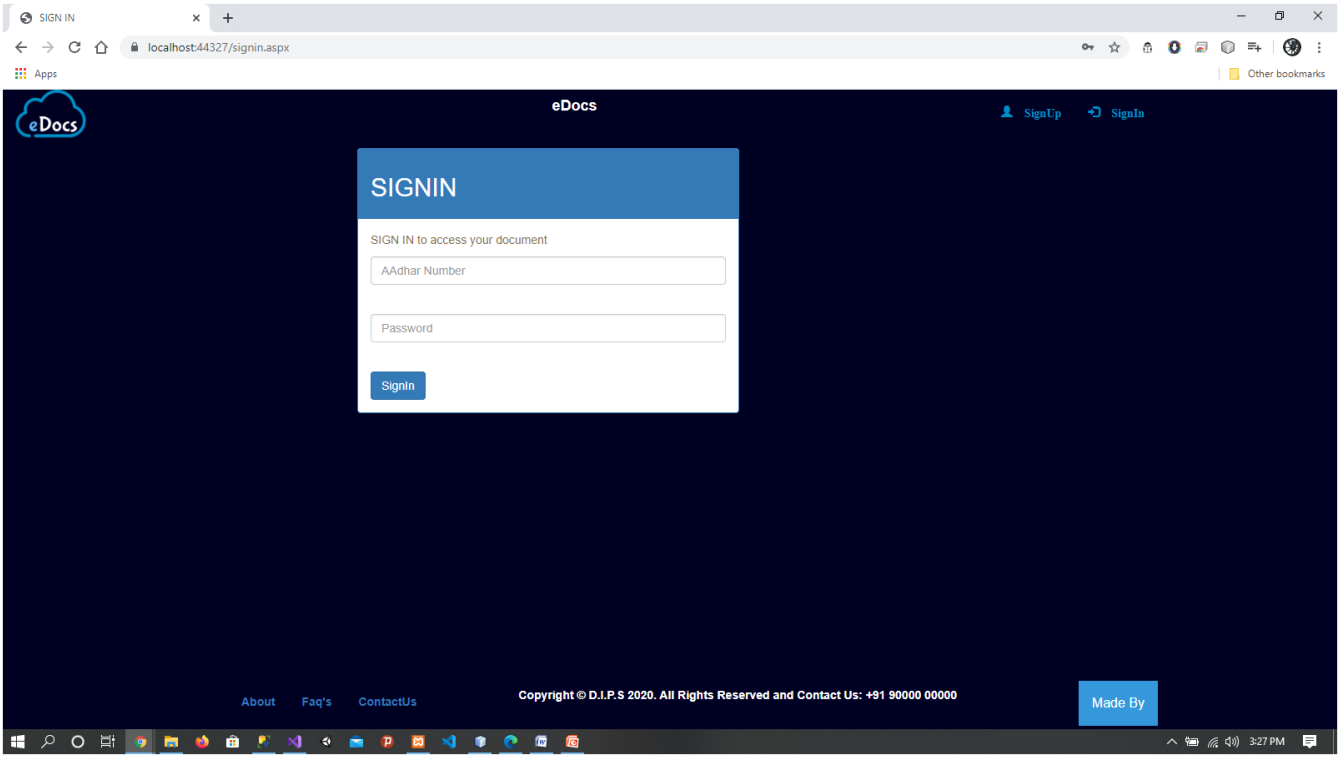
✔ Home page user interface:




✔ Login page user interface:




✔ Sign-up page user interface:




✔ Sign-In page user interface

## 3.3 Performance Requirements

- User Satisfaction: - The system is such that it stands up to the user expectations.
- Response Time: -The response of all the operation is good. This has been made possible by careful programming.
- Error Handling: - Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- Safety and Robustness: - The system is able to avoid or tackle disastrous action. In other words, it should be foul proof. The system safeguards against undesired events, without human intervention.
- Portable: - The software should not be architecture specific. It should be easily transferable to other platforms if needed.
- User friendliness: - The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

## 3.4 Design Constraints

✓ There are a number of factors in the client's environment that may restrict the choices of a designer. Such factors include standards that must be followed, resource limits, operating environment, reliability and security requirements and policies that may have an impact on the design of the system. An SRS (Software Requirements Analysis and Specification) should identify and specify all such constraints.

### 3.4.2 Hardware Limitation

✓ Minimum mobile hardware requirements
- Quad core Processor
- 1 GB RAM
- 2 GB memory
- 512kb Cache memory
-

✓ Minimum computer hardware requirements
- Processor :CORE i5
- Hard Disk :126 GB
- RAM :8 GB

## 3.5 Software System Attributes

✓ Computer software requirements
- Operating System :Windows XP/7/8/10
- Database :MYSQL
- Database connectivity :JDBC, MySQLi, ADO.NET
- Server :Apache Tomcat 5/6 Or GlassFish Server 4.1
- IDE :Eclipse or NetBeans
- Server Side/Scripting :Java, JSP, JavaScript, PHP, C#, ASP.Net
- RAM :8 GB

✓ Mobile software requirements

- Android OS above jelly bean

### 3.5.1 Reliability

- The capability to maintain the specified level of performance is called reliability.
- Unauthorized person will not be able to access the details.
- If any error is occur in-between running of program or software Then some recovery system is there which will backup our data , so even if the system crashes then to we will be able to recover data.
- There will be save points and roll backs for recovery of system.

### 3.5.2 Availability

- This system must be readily available to users who need to access their documents anytime.(24*7)
- The system must work relatively fast and must provide the data on request as soon as possible without affecting the quality and accuracy.

### 3.5.3 Security

- This system must be highly secured and must authenticate users strictly.
- This system would require handling confidential data. This system must keep the documents with more security as they will be stored in user's private account after retrieving.

### 3.5.4 Maintainability

- A commercial database is used for maintaining the database and the application server takes care of the site.
- In case of a failure, a re-initialization of the project will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.
- Maintenance is typically done after the software development has been completed. As the time evolves, so do the requirements and needs. It revolves around the understanding of the existing software and the effects of the change

### 3.5.5 Portability

- Portability is the ability of the system or application that can run in various environments. As the web application is based on the java, C#, PHP language, the application is portable

✓ Quality characteristics :

- Correctness - extent to which program satisfies specifications, fulfills user's mission objectives
- Efficiency - amount of computing resources and code required to perform function
- Flexibility - effort needed to modify operational program
- Interoperability - effort needed to couple one system with another
- Reliability - extent to which program performs with required precision

- Reusability - extent to which it can be reused in another application
- Testability - effort needed to test to ensure performs as intended
- Usability - effort required to learn, operate, prepare input, and interpret output

## 3.6 Other Requirements

### 3.6.1 Logical Database Requirements

- This section specifies the logical requirements for any information that is to be placed into a database. This may include:

**(1) User information:**
- Name
- Mobile Number
- Aadhar card Unique Id Number
- Date Of Birth
- Password

**(2) Admin information:**
- Mobile Number
- Password

**(3) Aadhar card Information**
- Aadhar card Unique Id Number
- Name
- Address
- Gender
- Date Of Birth
- Aadhar Document Image

**(4) Pan Card Information**
- Pan card Unique Id Number
- Aadhar card Unique Id Number
- Mobile number
- Pan Document Image

**(5) Voter-Id card Information**
- Voter-Id card Unique Id Number
- Aadhar card Unique Id Number
- Mobile number
- Voter Document Image

**(6) Driving license Information**
- Driving license Unique Id Number
- Aadhar card Unique Id Number
- Mobile number
- Date Of Issue
- Expiry Date
- Driving license Document Image

## 4. Appendices

The Appendices are not always considered part of the actual requirements specification and are not always necessary.

# Practical Statement:

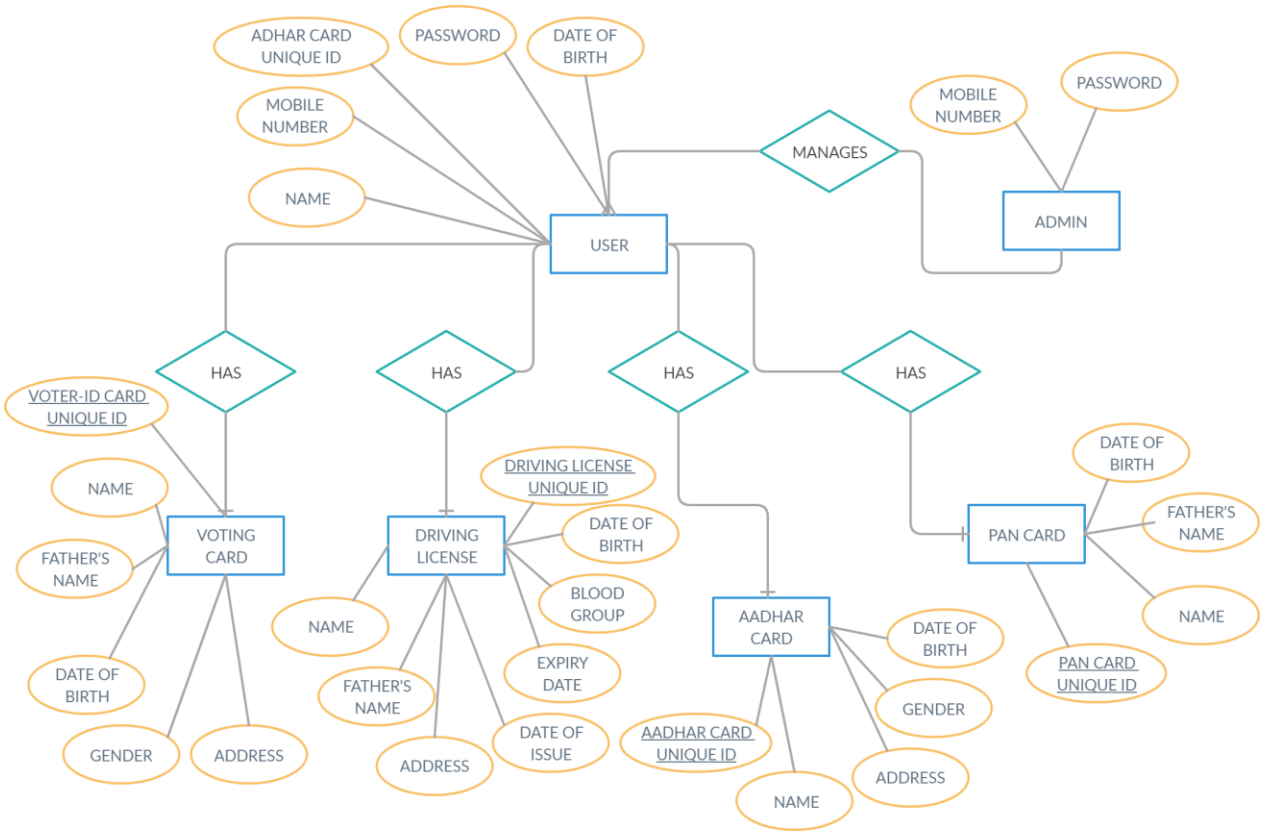
## 3. Design Analysis

Design Analysis represents and indicates the project specific flow and implementation by using different diagrams shown below

# Solution:

## 3.1 Draw E-R Diagram for your system.

ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong. Business information systems: The diagrams are used to design or analyze relational databases used in business processes.

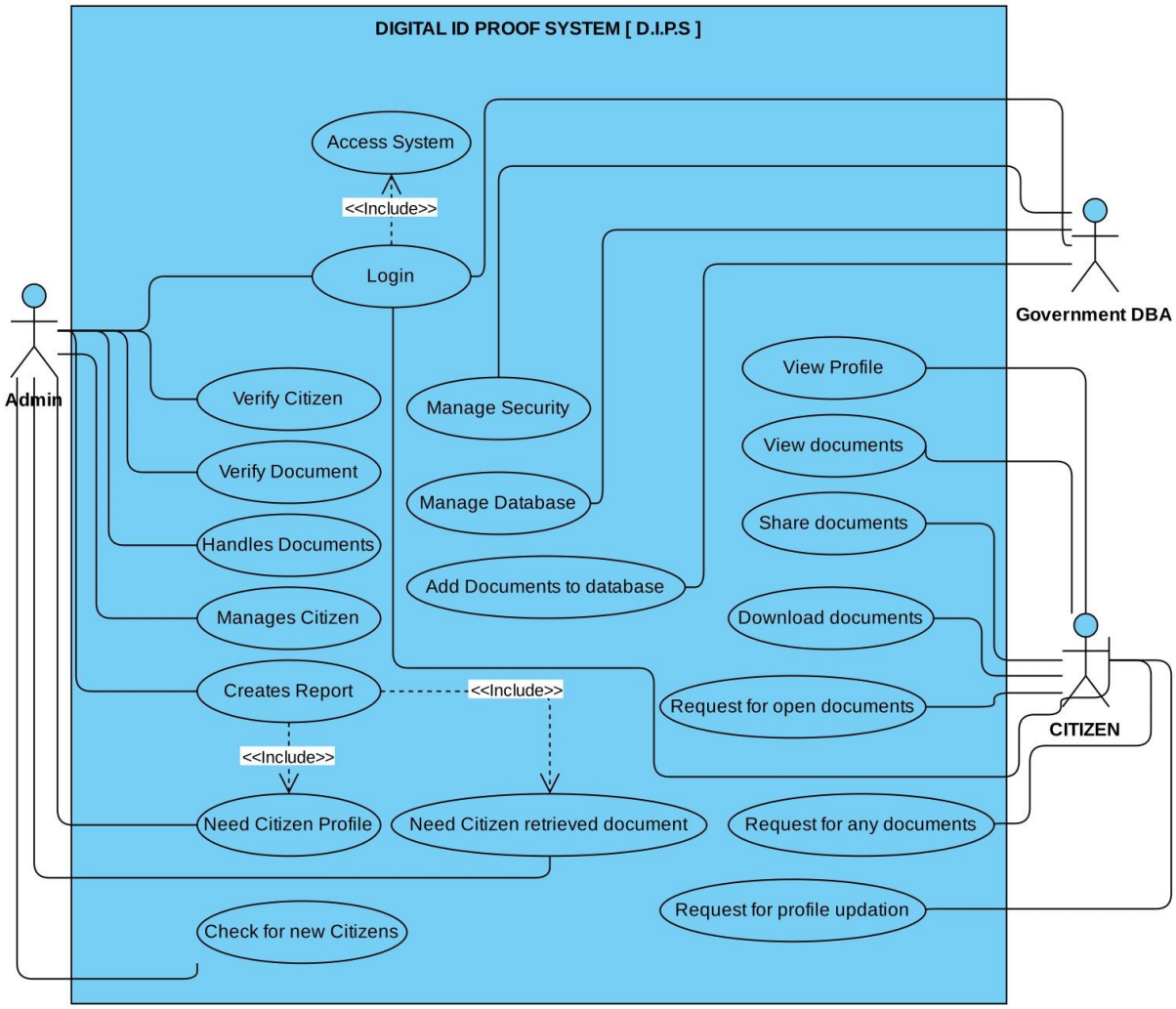


## 3.2 Design UML Diagrams

UML can be used to develop diagrams and provide users (programmers) with ready-to-use, expressive modelling examples. Some UML tools generate program language code from UML. ... UML is a graphical language for visualizing, specifying, constructing, and documenting information about software-intensive systems. It consist of other diagram that are show below.

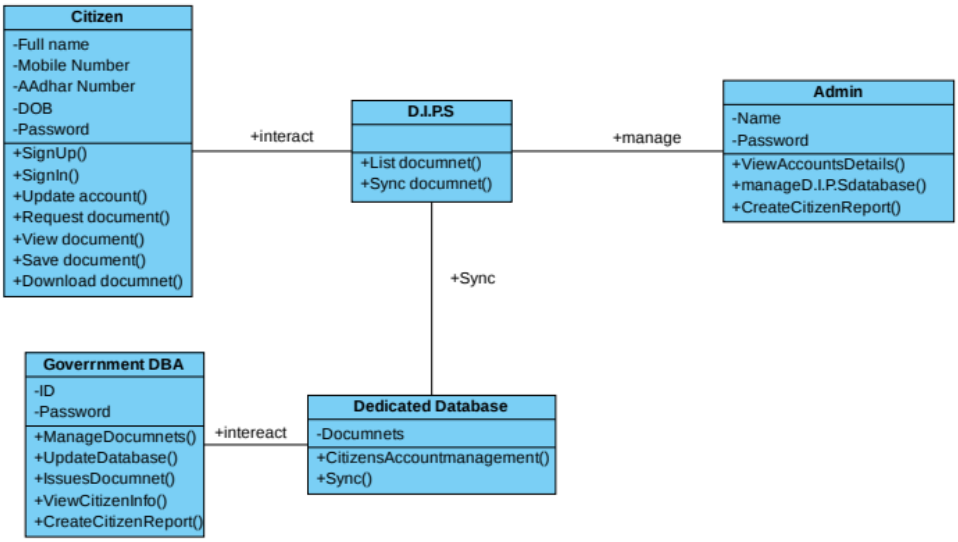
### 3.2.1 Use Case Diagrams

Use case diagrams are a way to capture the system's functionality and requirements in UML diagrams. It captures the dynamic behavior of a live system. A use case diagram consists of a use case and an actor. ... A purpose of use case diagram is to capture the core functionalities of a system.



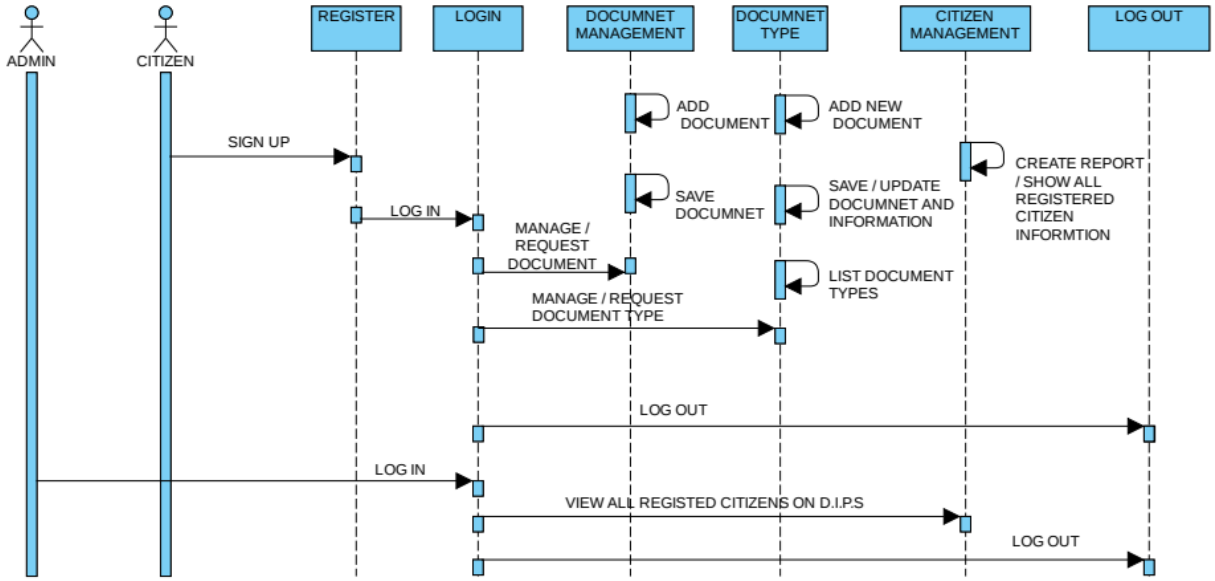
### 3.2.2 Class Diagram and Sequence Diagram

Class Diagram : Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.



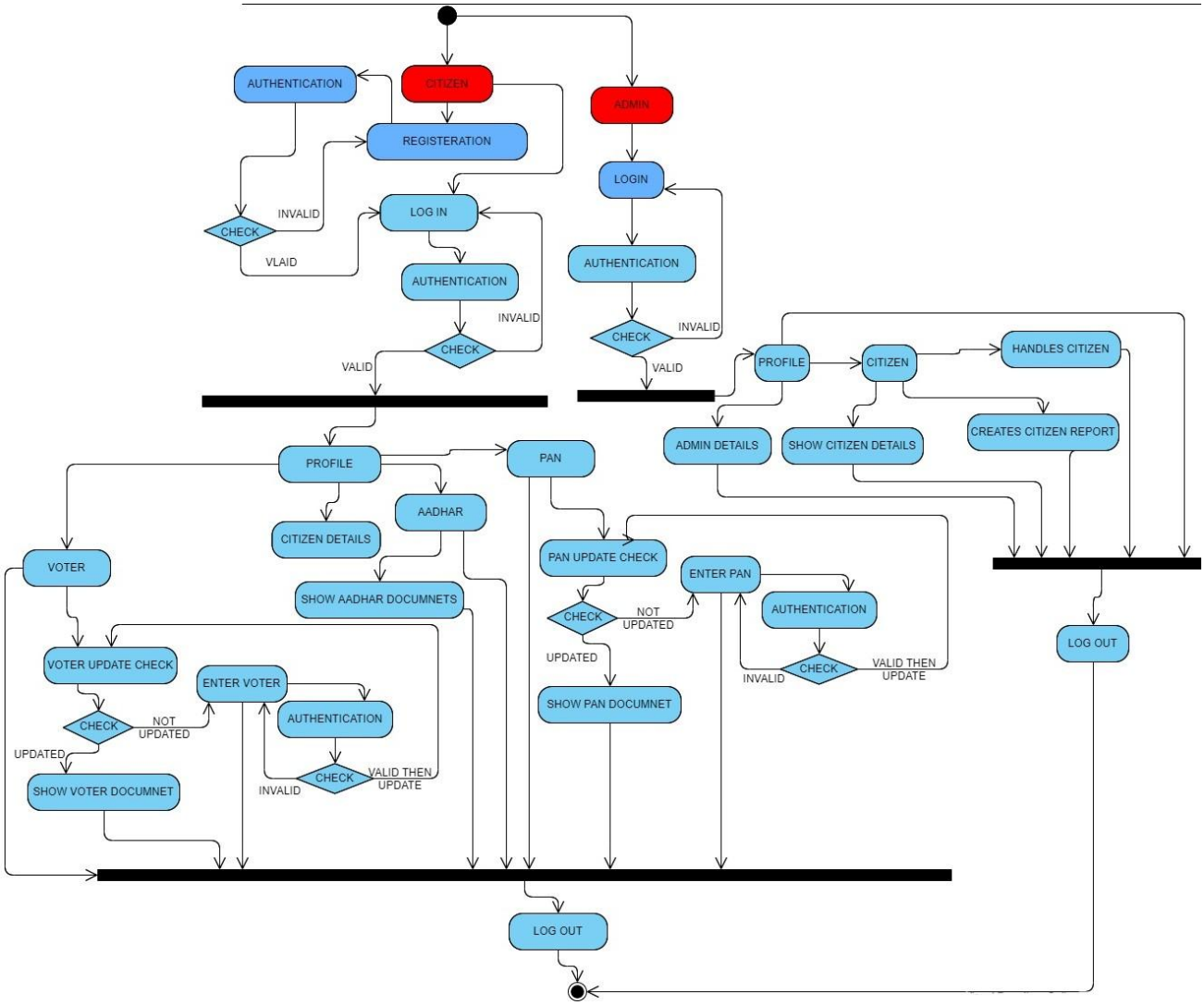
Sequence Diagram : A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.

These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

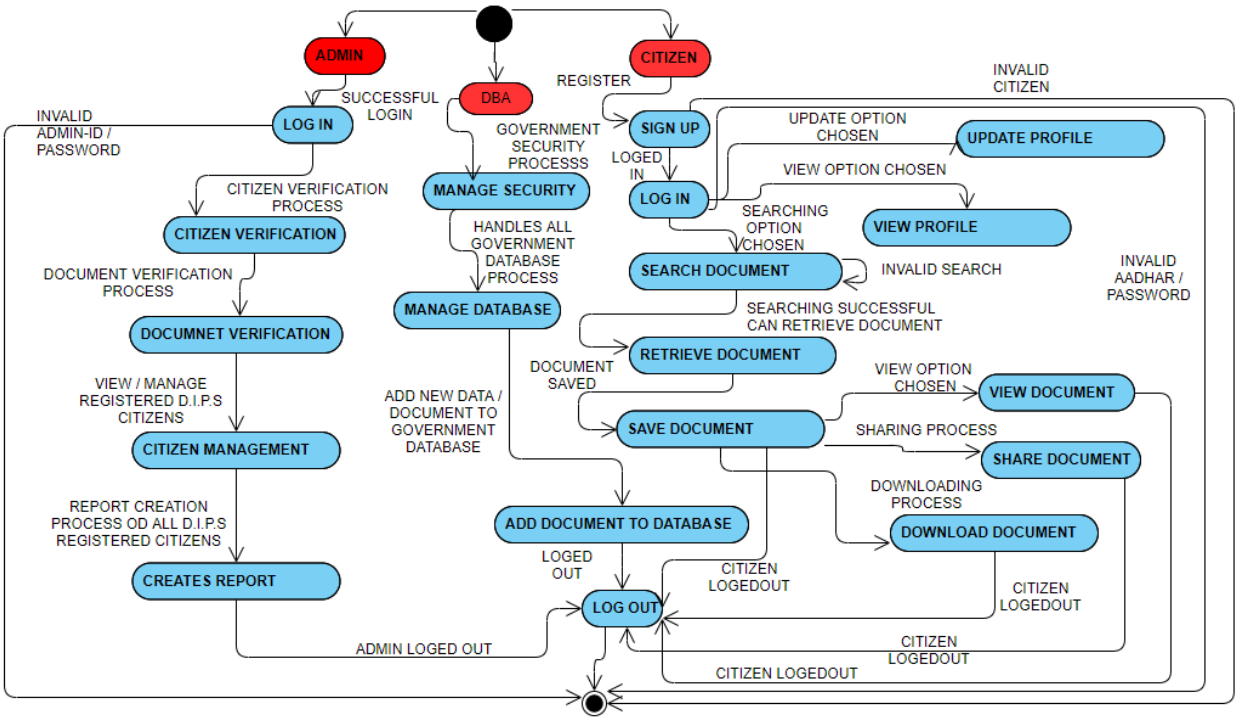


### 3.2.3 Activity Diagram and State Diagram

Activity Diagram : An activity diagram is used by developers to understand the flow of programs on a high level. It also enables them to figure out constraints and conditions that cause particular events. A flow chart converges into being an activity diagram if complex decisions are being made.



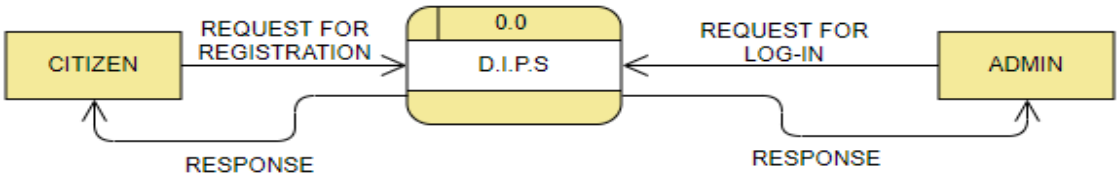
State Diagram : A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams.



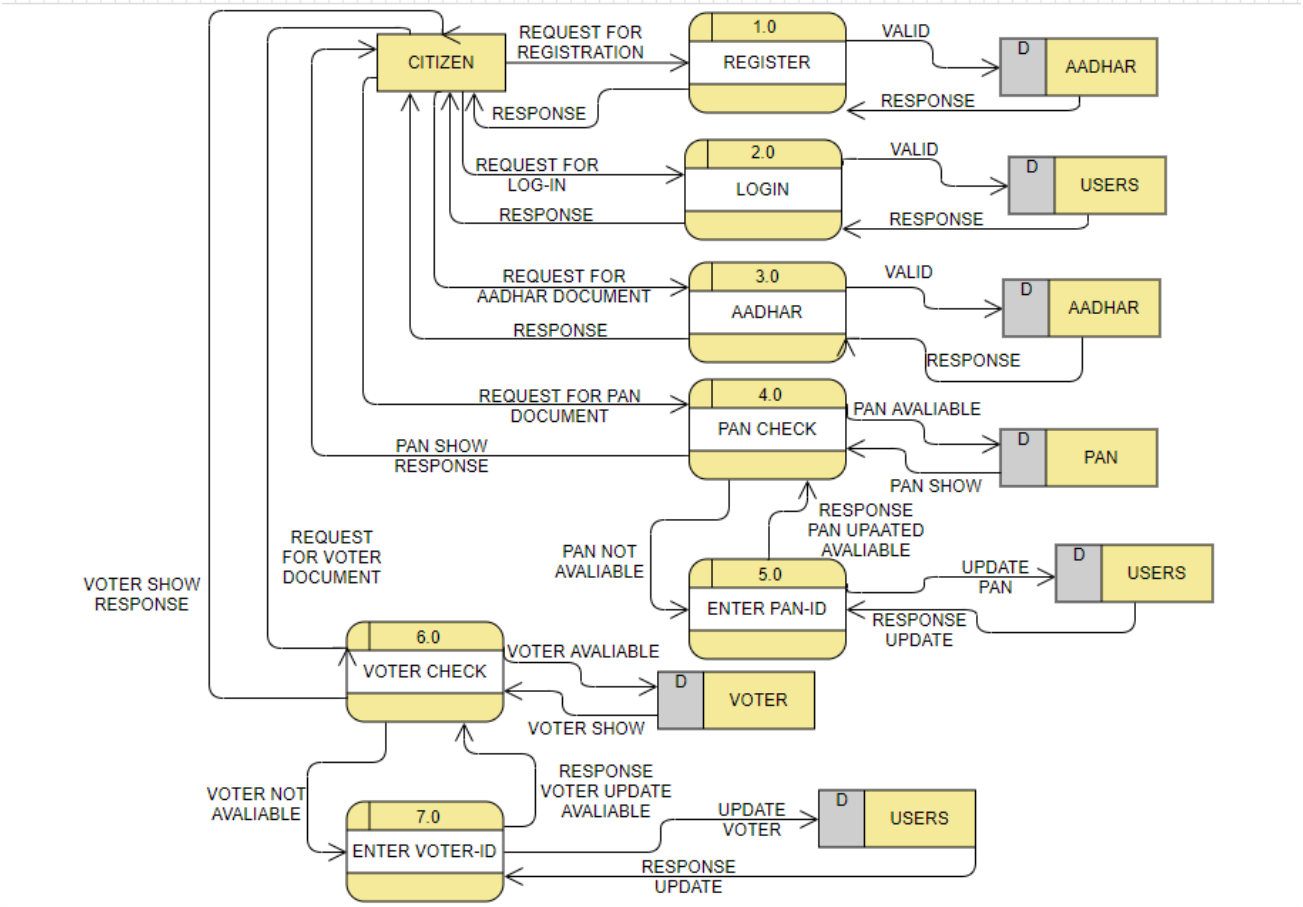
## 3.3 Design DFD level -0 and level-1 for your system

DFD : Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical.

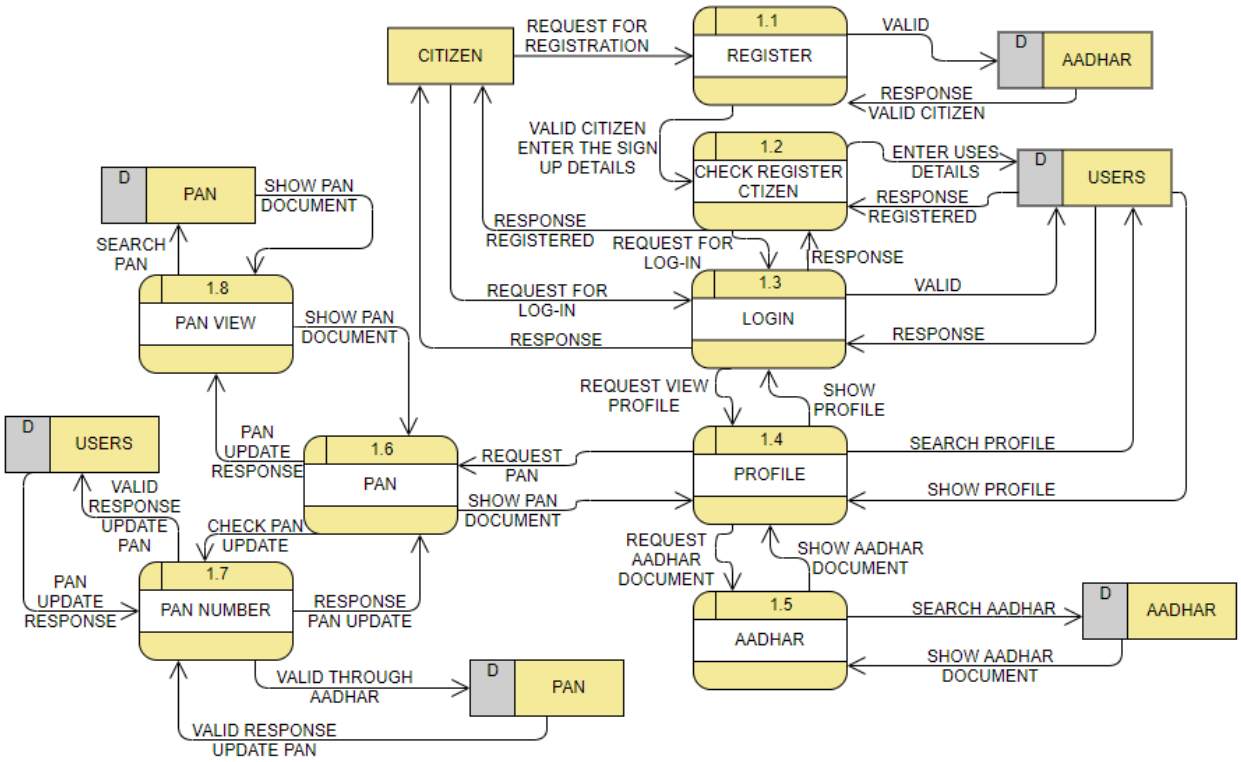
DFD LEVEL “0” :



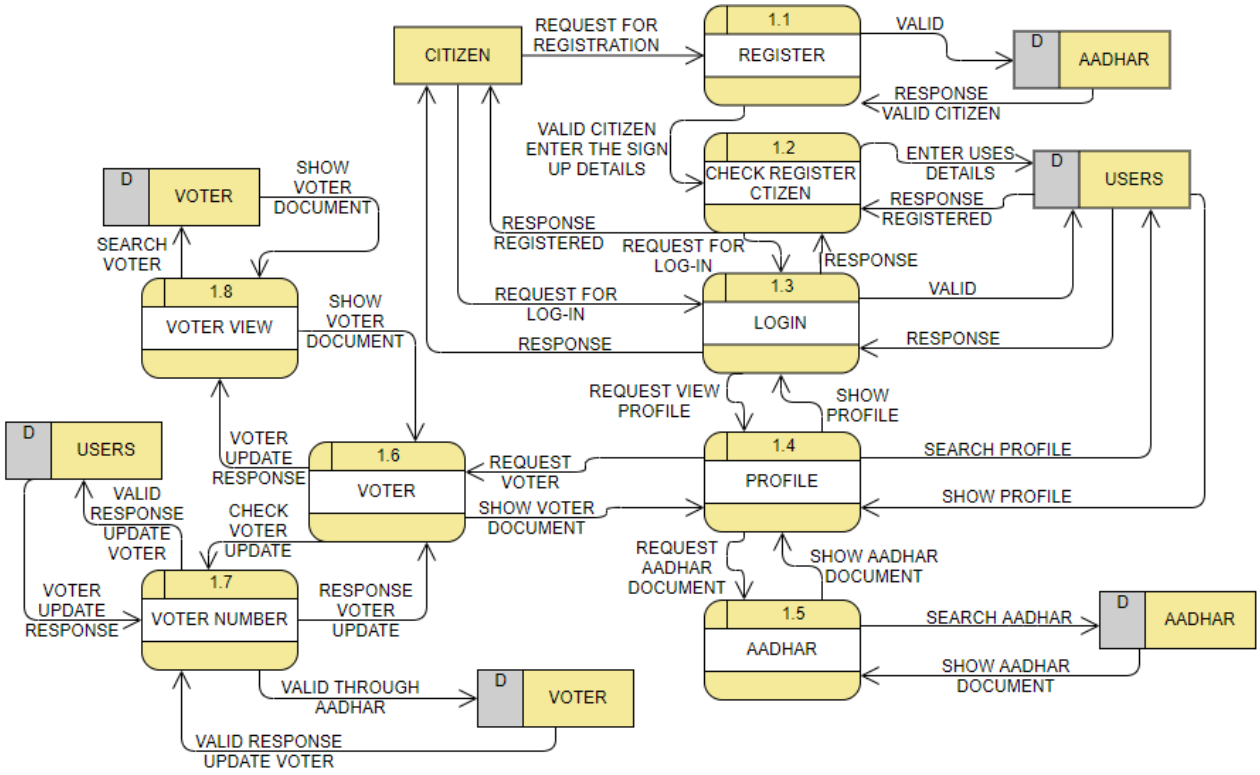

DFD LEVEL “1” CITIZEN :



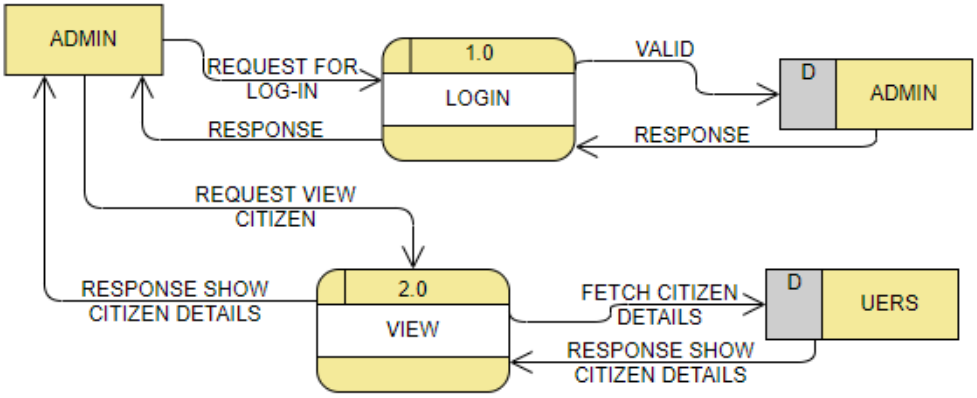
DFD LEVEL “2” CITIZEN-PAN DOCUMENT :



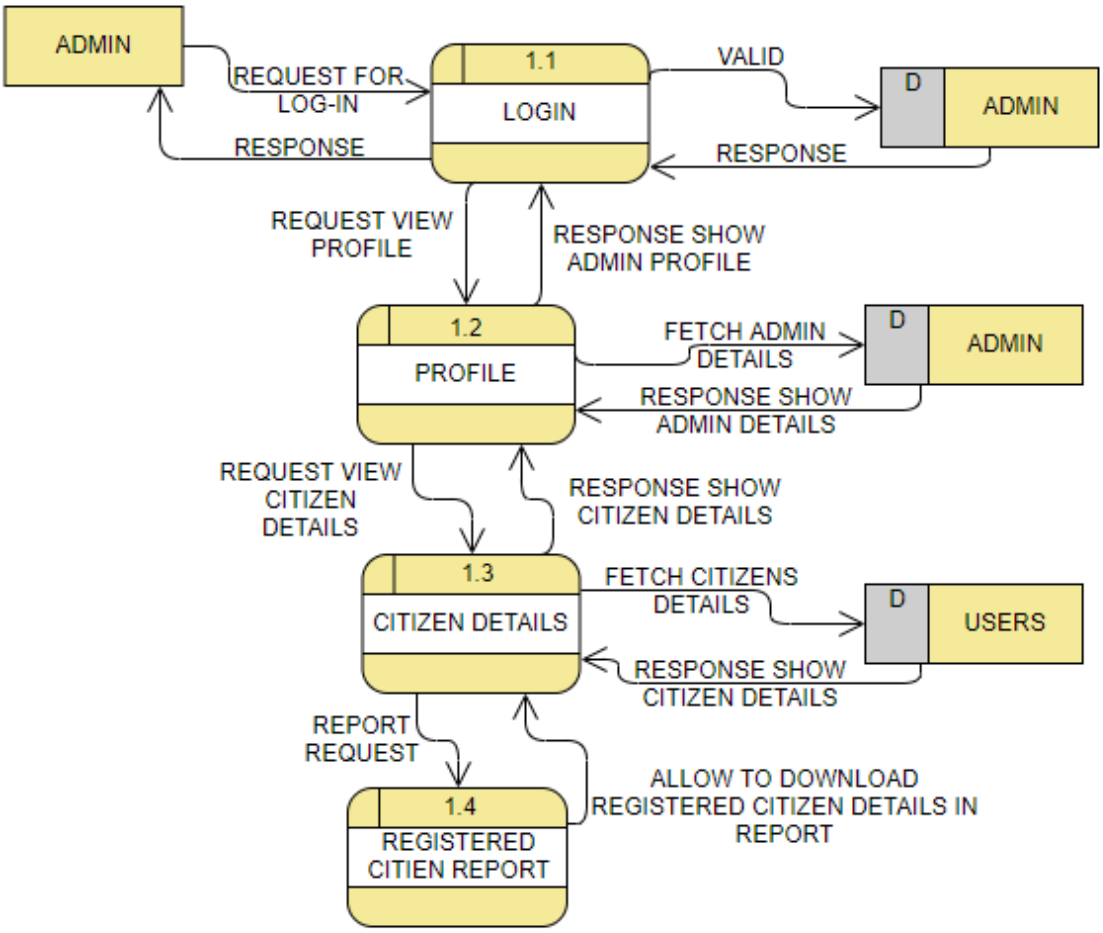

DFD LEVEL “2” CITIZEN-VOTER DOCUMENT :



DFD LEVEL “1” ADMIN :

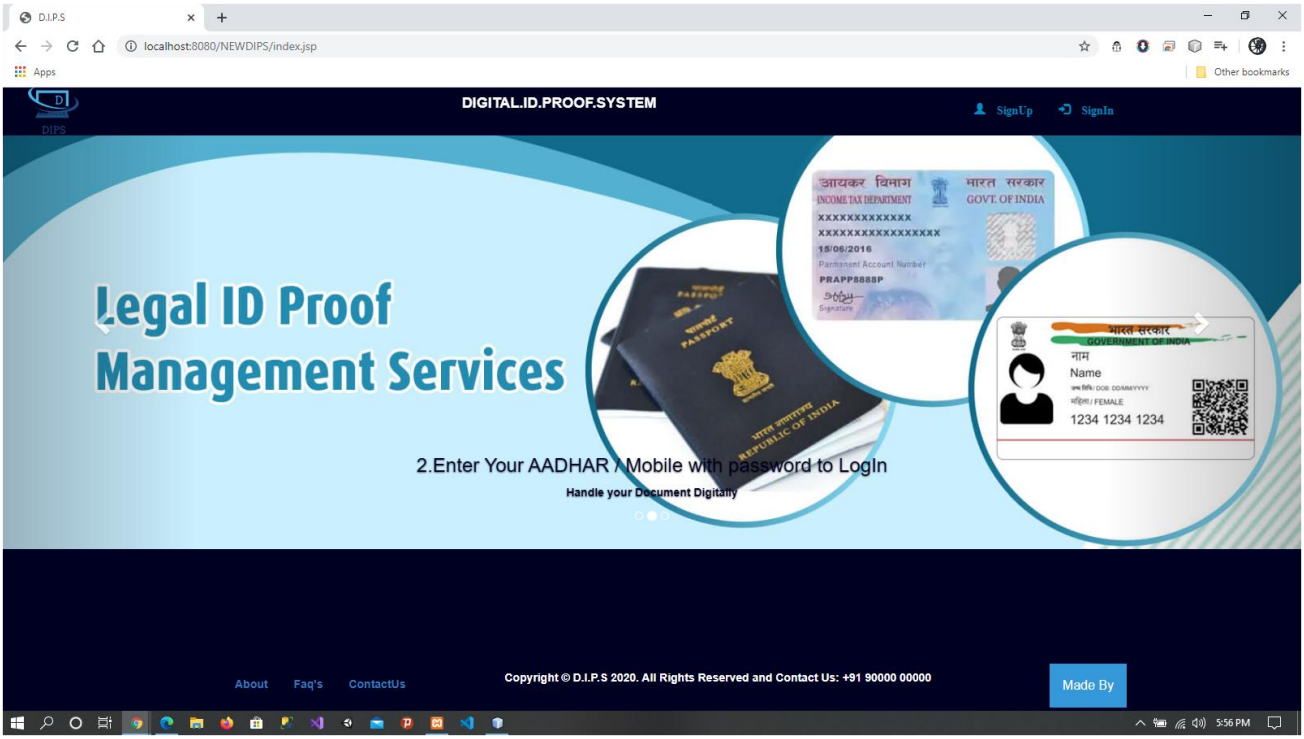



DFD LEVEL “2” ADMIN :

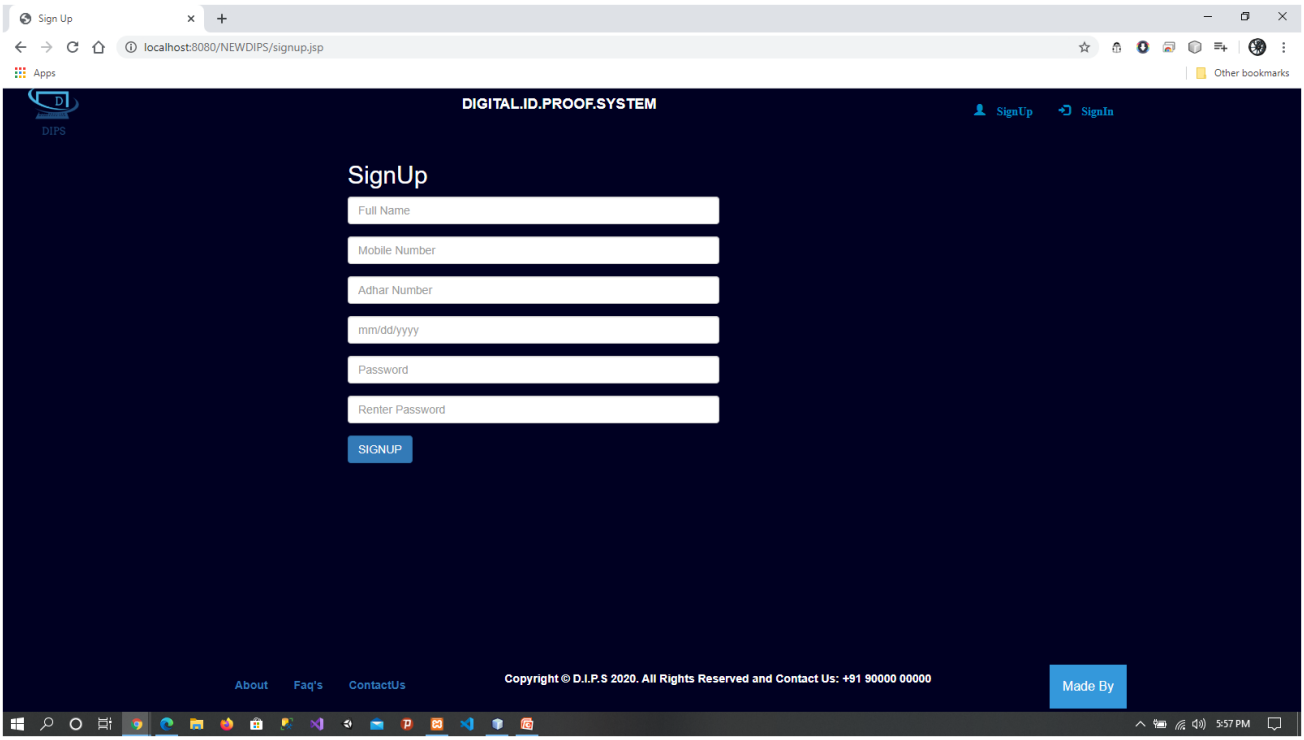


3.4 Prepare User interface design (Front end) and also mention characteristics of it :

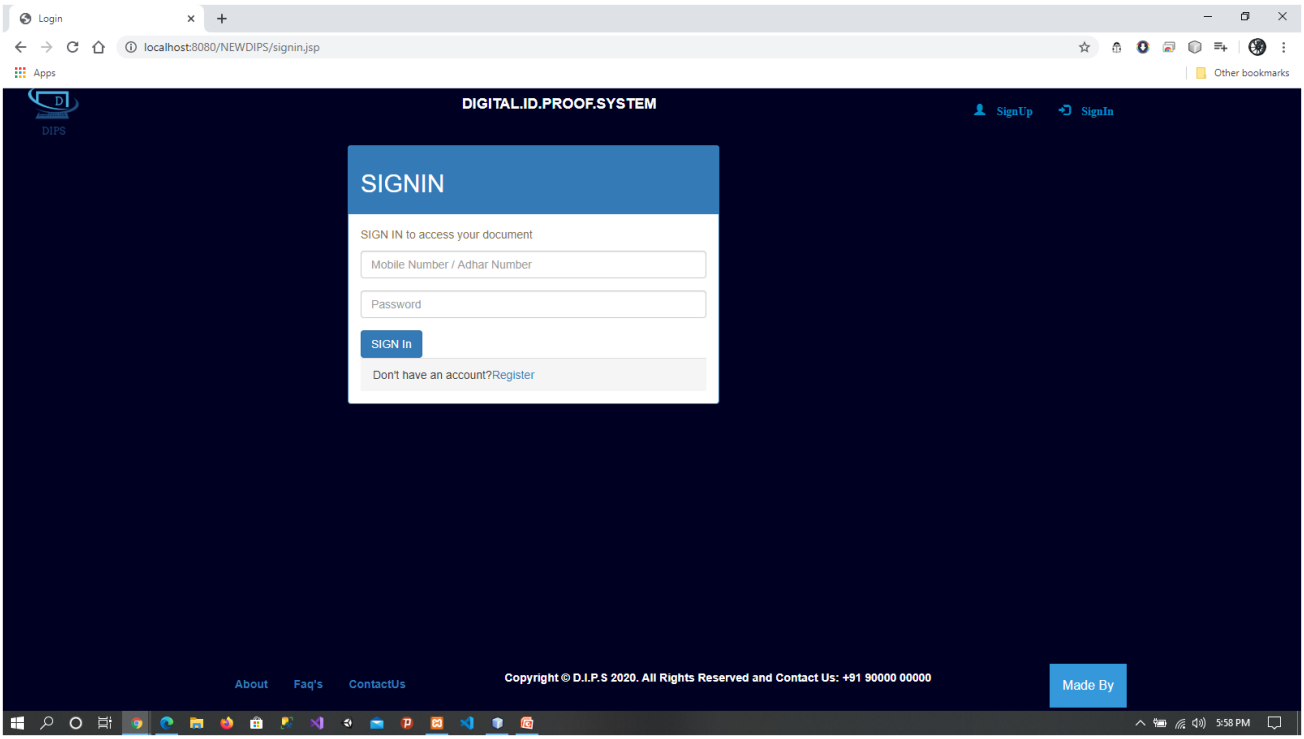
USER INTERFACE – INDEX PAGE :




USER INTERFACE – SIGN-UP PAGE :



## USER INTERFACE – SIGN-IN PAGE :




## CHARECTRICITS OF USER INTERFACE :

### INDEX PAGE :

- Citizens can have information of entire website and how to operate it.
- Citizens can have information of how to fetch the document and save it to theirs profile.
- Citizens can have faq's to solve their problems.

### SIGN UP PAGE :

- Citizens have to be valid "AADHAR" holder to register.
- Citizens have to enter the few specific details to get successfully registered of D.I.P.S .
- Citizens need to give there full name, Aadhar number, mobile number, Date-of-birth, and specific limited length password

### SIGN IN PAGE :

- Citizen can login in with Aadhar number and mobile number.

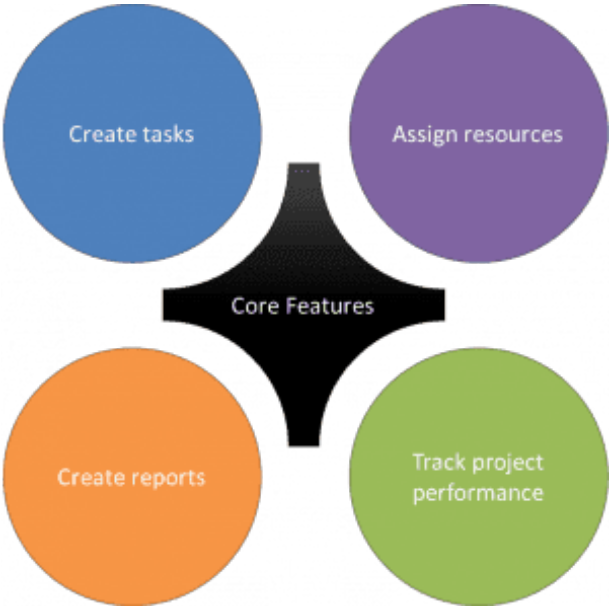
# 4. Software Project Management

## Examine Project Management Tool and prepare a report.

If you are managing a project that needs a detailed schedule, Microsoft Project is an excellent tool to use. In this tutorial, I am going to show you the Microsoft Project basics to build a project schedule. This tutorial works with Microsoft Project 2013 or 2016 (Standard, Pro or from Office 365).

### What is Microsoft Project?

Microsoft Project is a desktop application you install on your computer. With Microsoft Project, you can build and track project schedules. As you can imagine, there are essential features for the beginner and more advanced features for broad and complex projects. Microsoft Project offers four core features as shown in the following figure.



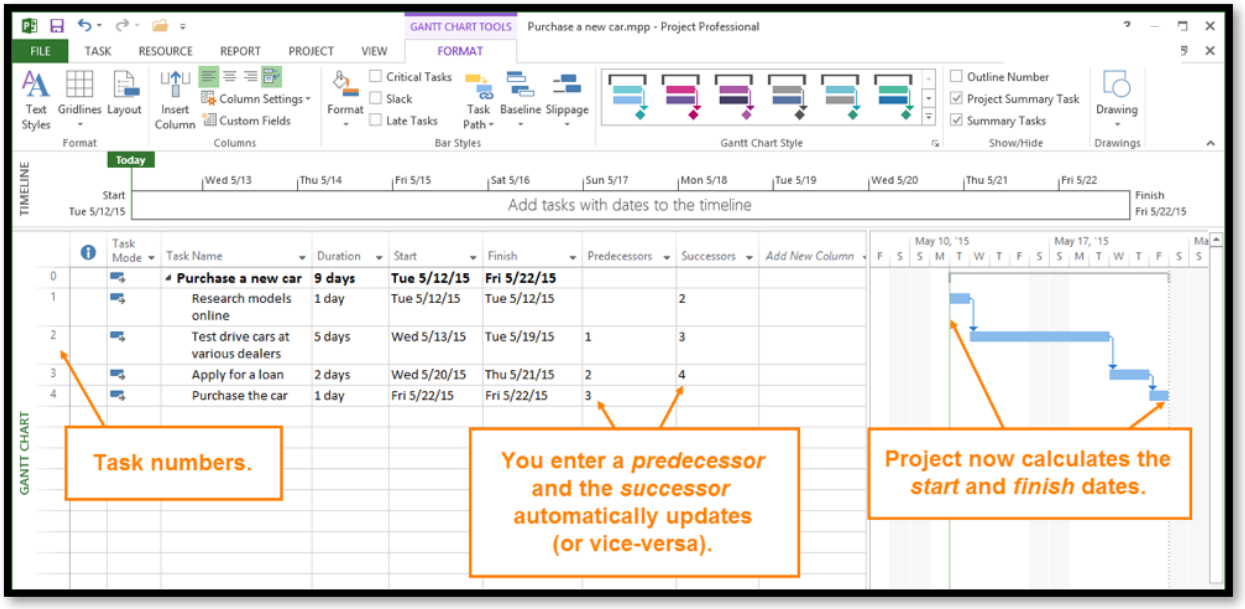

Core Microsoft Project features

- **Create tasks** that represent the steps to complete the project.
- **Assign resources** such as the people, materials, or equipment needed to deliver the project.
- **Track project performance** by comparing the project's current state of completion to the original baseline plan.
- **Create reports** to visualize the project's progress and share them with your project team, stakeholders, and sponsors.

For this tutorial, I am going to show you how to create tasks and link them together using constraint-based scheduling.

## Constraint-based scheduling

You could probably use a spreadsheet to list out a bunch of tasks and enter some dates. However, if you change the date of one task and want that to roll down and adjust the dates of other tasks, you are getting into some complicated math and scripting. Microsoft Project helps you automatically calculate the start and finish dates of tasks by using constraints called *predecessors* and *successors*, as shown in the following figure.




By constraining tasks with a predecessor/successor relationship, the tasks automatically calculate.

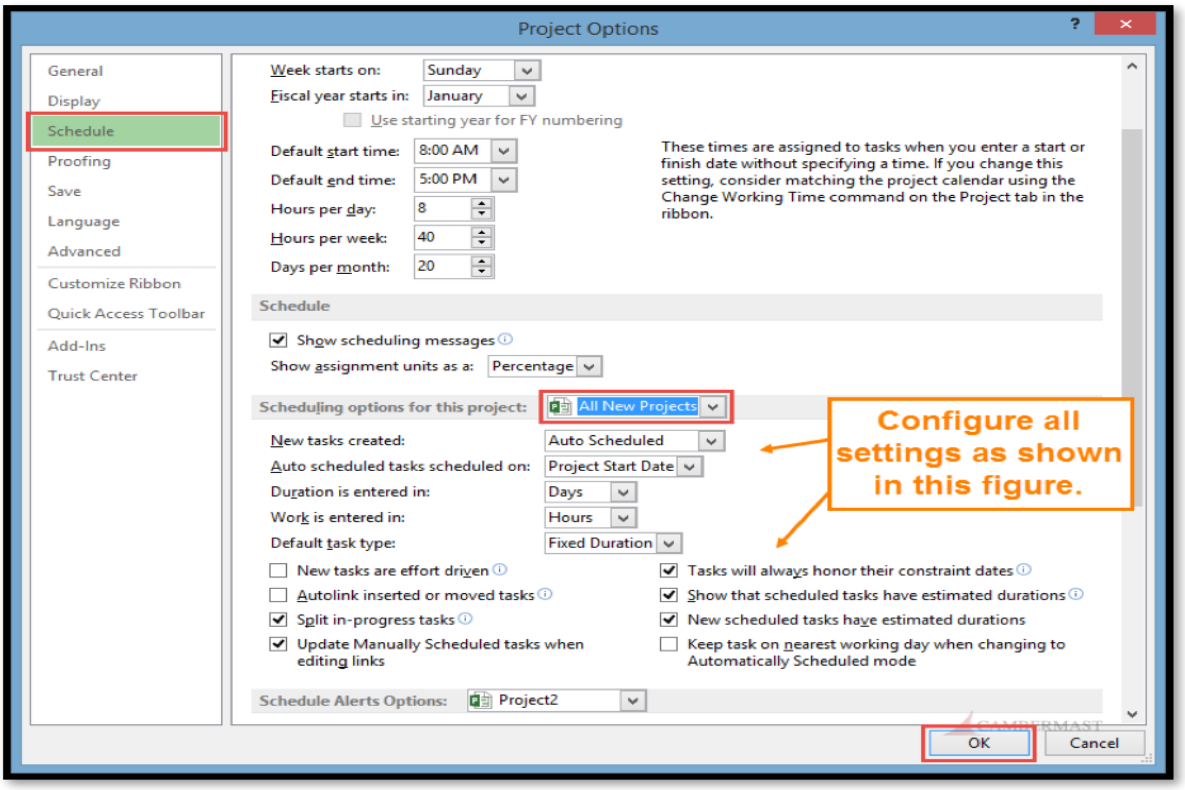
As you can see in that previous figure, task #1 (*research models online*) starts before task #2 (*test drive cars at various dealers*). You represent this by typing the number ***1*** in line 2's *predecessor* column. The graphical chart on the right is called a *Gantt* chart. The Gantt chart shows how Microsoft Project is automatically calculating when one task should start and the other should finish.
This task-linking feature alone is what makes Microsoft Project so compelling to project managers.

## Activity: Run and configure Microsoft Project for the first time

In this activity, you will configure settings based on my recommendations. Given that this tutorial is just to get you started, I am not going to explain each one of these settings, but rest assured these are generically the best practices that most project managers follow.

1. Run Microsoft Project and open a ***blank*** project.
2. Click the ***File*** tab and then select the ***Options*** menu item.
3. The *project options* dialog appears as shown in the figure below.

- Select the ***Schedule*** item.
- Locate the *scheduling options for this project* section and then select the ***All New Projects*** item from the pick list.
- *New tasks created*: ***Auto Scheduled***.
- *Auto scheduled tasks scheduled on*: ***Project Start Date***.
- *Duration is entered in*: ***Days***.
- *Work is entered in*: ***Hours***.
- *Default task type*: ***Fixed Duration***.
- *New tasks are effort driven*: ***deselect***.
- *Autolink inserted or moved tasks*: ***deselect***.
- *Keep task on nearest working day when changing to Automatically Scheduled mode:* ***deselect.***
- *All other checkboxes in the scheduling options section:* ***select***.
- Click the ***OK*** button.



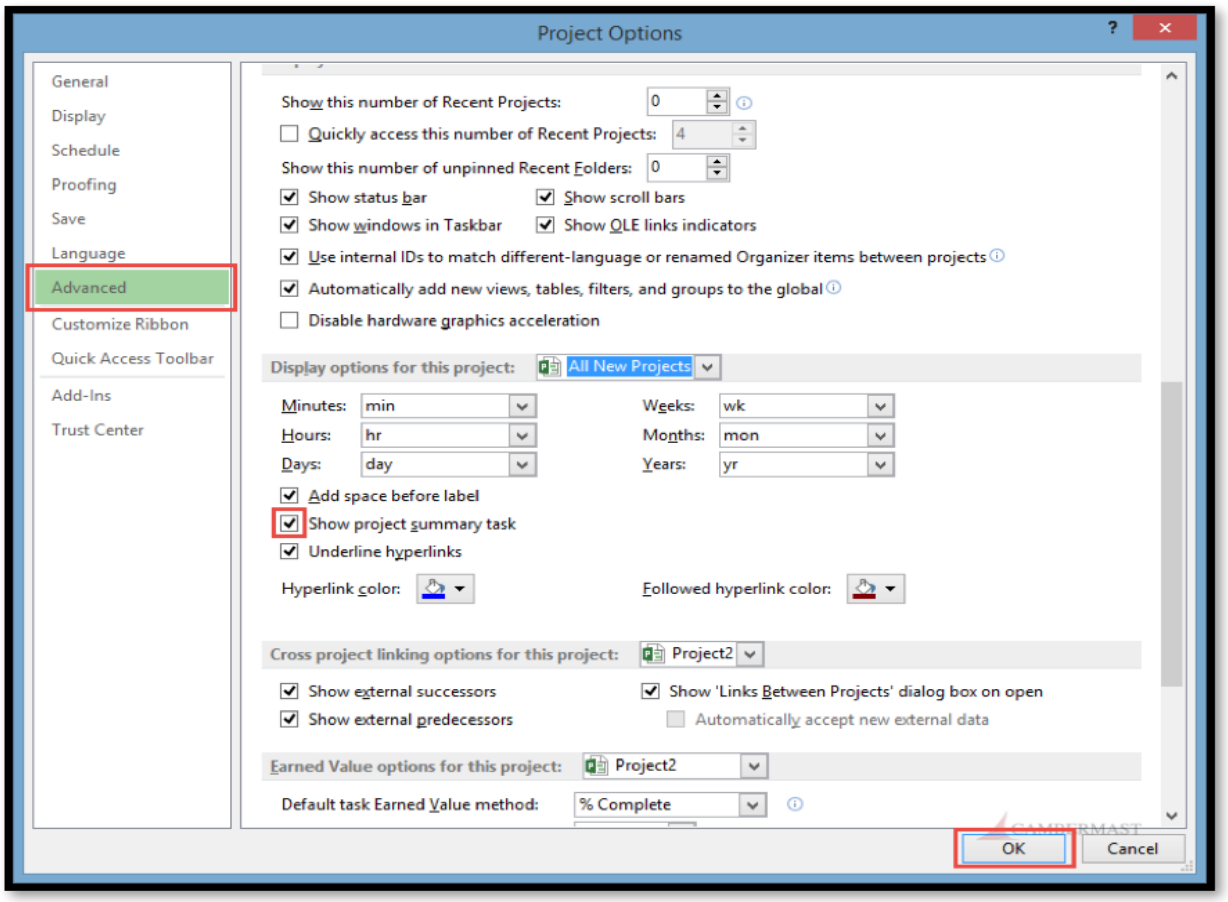

The project options dialog with best practice scheduling options.

## Activity: Enable the project summary task

A common mistake people make in Microsoft Project is to create a task with the project name at the very top of their project plan. Microsoft Project will do this for you automatically by displaying the *project summary task*.
Since the *project summary task* is not a default option, you need to enable it. Follow these steps to display the *project summary task* at the top of your project:

1. Click the ***File*** tab and then select the ***Options*** menu item.
2. The *project options* dialog appears as shown in the following figure.
3. Select the ***Advanced*** item.

- Locate the *Display options for this project* section and select ***All New projects*** from the pick list.
- Select the ***Show project summary task***.
- Click the ***OK*** button.




The project options dialog with the show project summary task option selected.

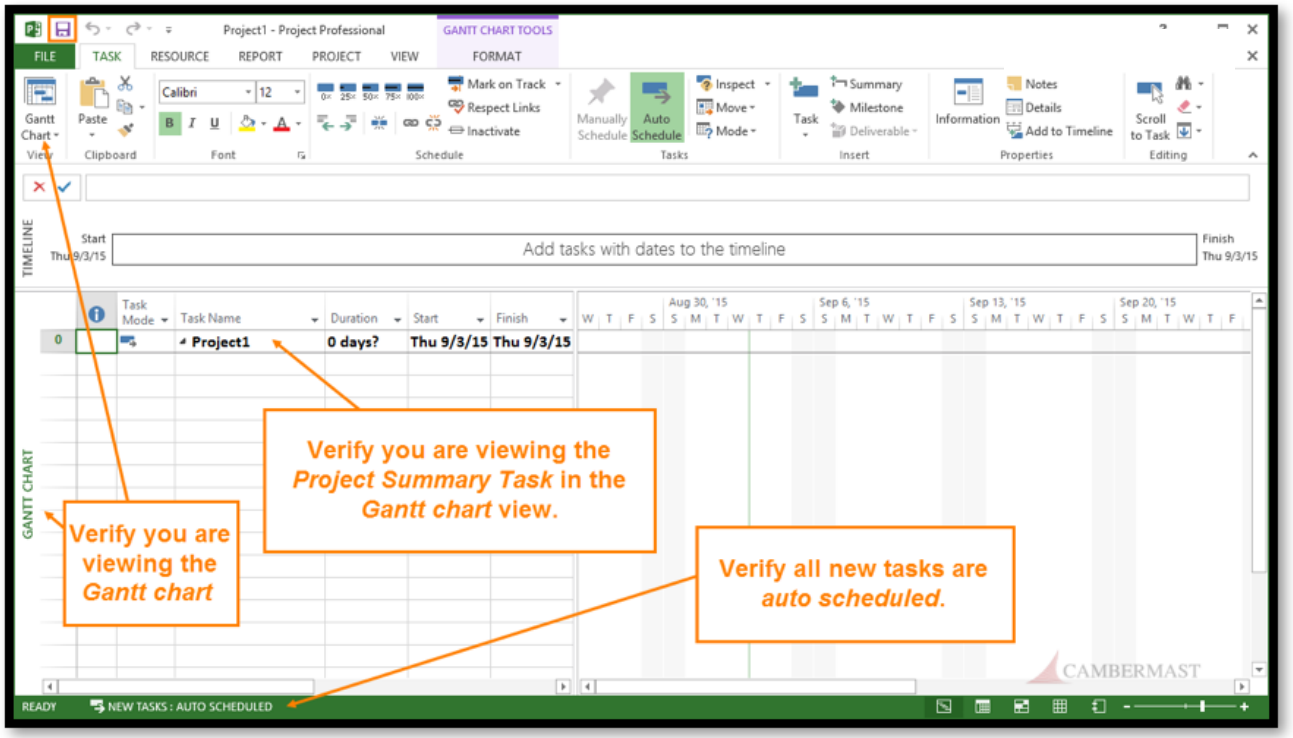
From now on, when you create a project, the project summary task will display (task zero).

## Activity: Verify you are ready to create a project

At this point, you should be looking at an empty project plan. Follow these steps to make sure Microsoft Project is ready, so you can create a project:

1. The blank project should be open as shown in the following figure (the project name might be different for you).

- Click the ***Task*** tab and then click the ***Gantt chart*** icon.
- ***Verify*** you can view the *project summary task*.
- ***Verify*** all new tasks are *auto scheduled* by viewing the *status bar* at the bottom-left side of the application window.




A new blank project with all the verification steps.

## Activity: Save your project

Before you start typing tasks, save your project:

1. Click the ***File*** tab, select the ***Save As*** item, and save the file to a location of your choosing.
2. Name the file ***My New House***.

## Activity: Add some tasks

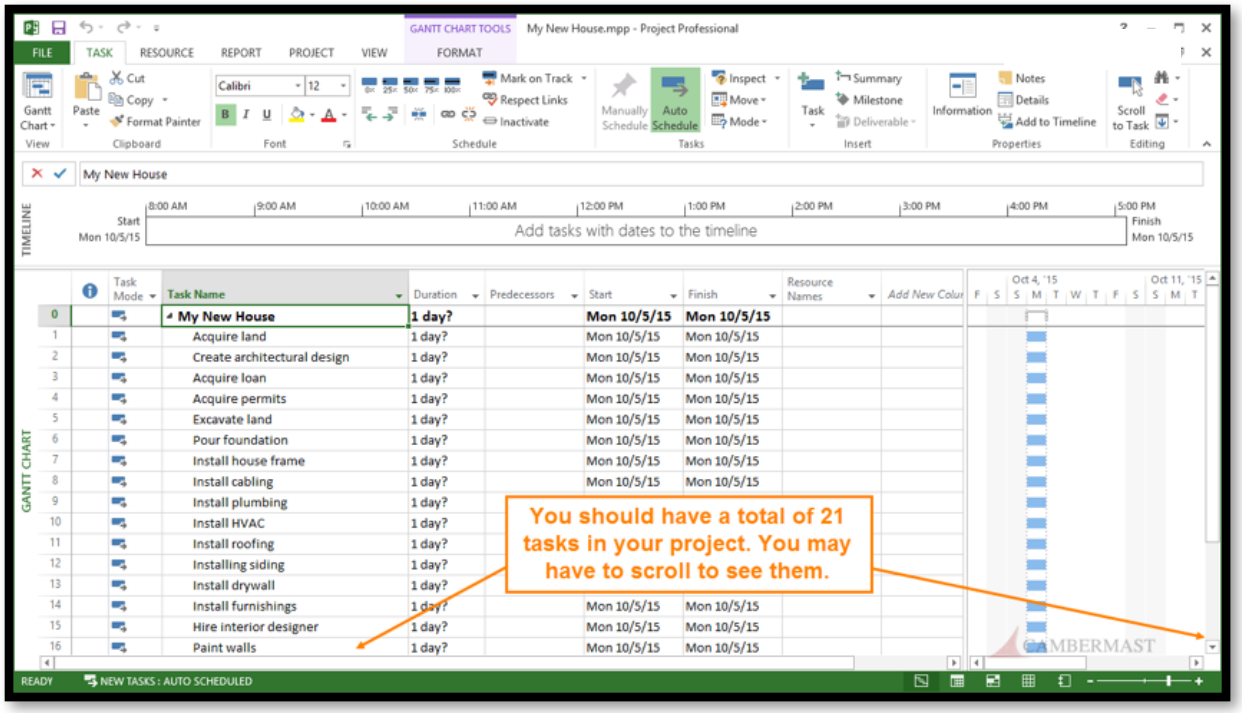
Using Microsoft Project is very similar to using Microsoft Excel. Click in the first *task name* cell and start typing.

1. Add the tasks in the following figure:

| ID | Task Name |
|---|---|
| 0 | **My New House** |
| 1 | Acquire land |
| 2 | Create architectural design |
| 3 | Acquire loan |
| 4 | Acquire permits |
| 5 | Excavate land |
| 6 | Pour foundation |
| 7 | Install house frame |
| 8 | Install cabling |
| 9 | Install plumbing |
| 10 | Install HVAC |
| 11 | Install roofing |
| 12 | Install siding |
| 13 | Install drywall |
| 14 | Install furnishings |
| 15 | Hire interior designer |
| 16 | Paint walls |
| 17 | Build out kitchen |
| 18 | Add furniture |
| 19 | Walkthroughs and final approvals |
| 20 | Pay final contractor fees |
| 21 | Move in |

The list of tasks to add to your project.

Your project should look similar the following figure:




The new project containing all the tasks.

## Activity: Estimate task durations

After you define the tasks in your project, you will estimate the durations. You can estimate durations in seconds, minutes, days, weeks, months, quarters, and years. I recommend you **always** use *days*. This way, it is easy for anyone to scan the project without mistaking a 2m (months) for 2d (days).

1. Locate the *duration* field and type the durations as shown in the following figures.

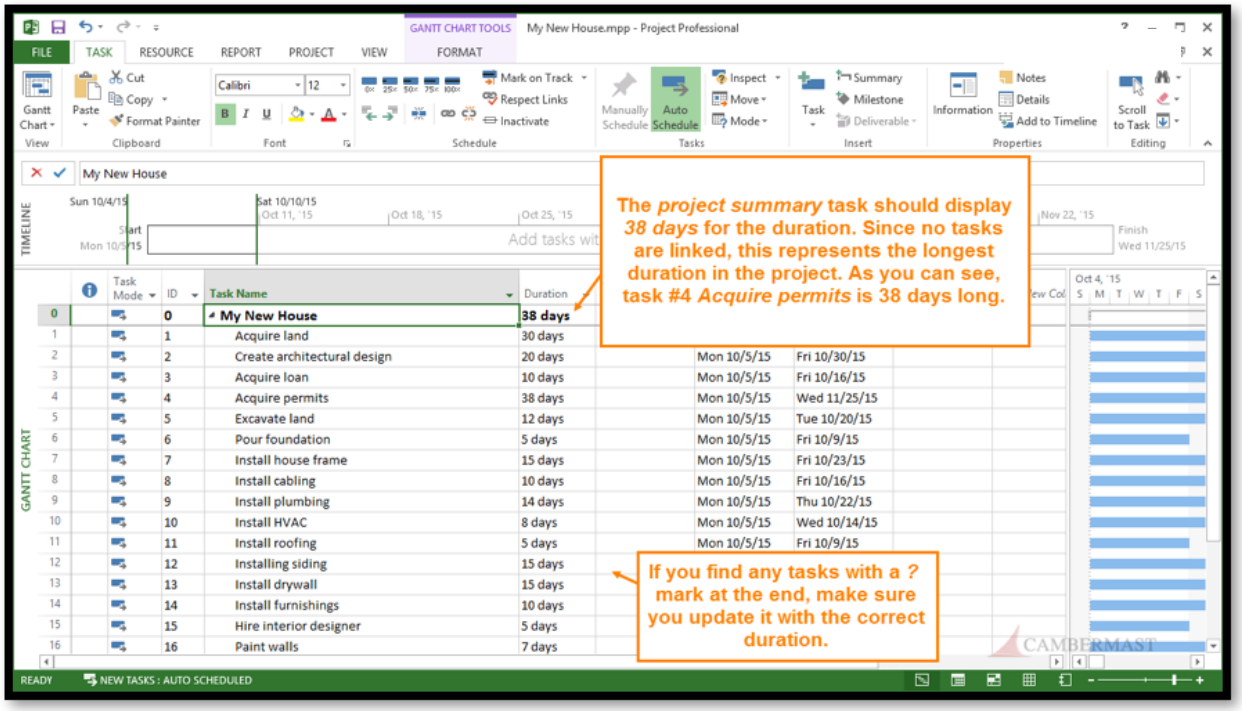
- **Note**: You do not have to type out the duration completely. You can just type ***30, 20, 10***, etc. and Microsoft Project will automatically add the *days* text. If there is a problem, type ***30d, 20d, 10d***, etc.

| ID | Task Name | Duration |
|---|---|---|
| 11 | Install roofing | 5 days |
| 12 | Install siding | 15 days |
| 13 | Install drywall | 15 days |
| 14 | Install furnishings | 10 days |
| 15 | Hire interior designer | 5 days |
| 16 | Paint walls | 7 days |
| 17 | Build out kitchen | 14 days |
| 18 | Add furniture | 3 days |
| 19 | Walkthroughs and final approvals | 5 days |
| 20 | Pay final contractor fees | 2 days |
| 21 | Move in | 0 days |

The durations to enter for each task.

| ID | Task Name | Duration |
|---|---|---|
| **0** | **My New House** | **38 days** |
| 1 | Acquire land | 30 days |
| 2 | Create architectural design | 20 days |
| 3 | Acquire loan | 10 days |
| 4 | Acquire permits | 38 days |
| 5 | Excavate land | 12 days |
| 6 | Pour foundation | 5 days |
| 7 | Install house frame | 15 days |
| 8 | Install cabling | 10 days |
| 9 | Install plumbing | 14 days |
| 10 | Install HVAC | 8 days |

Now your project plan should look something like the figure below:




Your project plan after entering durations.

## Activity: Link your tasks with predecessors

At this point, your project only shows a duration of 38 days. That is because all the tasks start on the same date, so Microsoft Project is calculating the longest task (38 days). Follow the steps below to link all the tasks together and allow Microsoft Project to figure out the start and end date for each task:

1. Locate the *Predecessors* column and enter the numbers shown in the figures below.

- **Note**: You can link one task to multiple tasks by separating each number with a comma.

| ID | Task Name | Predecessors |
|---|---|---|
| 11 | Install roofing | 10 |
| 12 | Install siding | 11 |
| 13 | Install drywall | 12 |
| 14 | Install furnishings | 13 |
| 15 | Hire interior designer | 7 |
| 16 | Paint walls | 13 |
| 17 | Build out kitchen | 16 |
| 18 | Add furniture | 17 |
| 19 | Walkthroughs and final approvals | 18 |
| 20 | Pay final contractor fees | 19 |
| 21 | Move in | 20 |

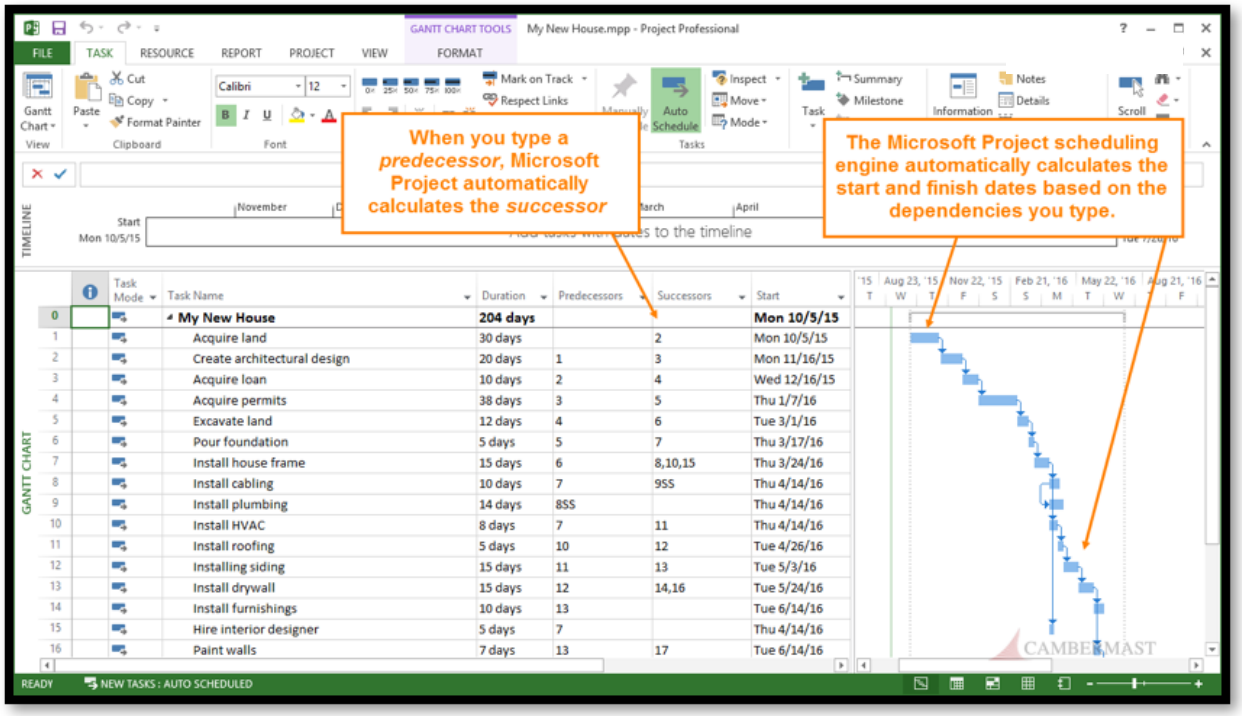
| ID | Task Name | Predecessors |
|---|---|---|
| 0 | **My New House** | |
| 1 | Acquire land | |
| 2 | Create architectural design | 1 |
| 3 | Acquire loan | 2 |
| 4 | Acquire permits | 3 |
| 5 | Excavate land | 4 |
| 6 | Pour foundation | 5 |
| 7 | Install house frame | 6 |
| 8 | Install cabling | 7 |
| 9 | Install plumbing | 8SS |
| 10 | Install HVAC | 7 |

Link all the tasks so Microsoft Project can calculate the start and finish date for your project

As you type predecessors, Microsoft Project automatically fills in the *successor* column. If you want to see the successors column, follow these steps:

1. ***Right-click*** the heading of the *start* column.
2. A menu appears. Select ***Insert Column***.
3. A list of available fields appears. Type the word ***Successor*** and then press the ***Enter*** key on your keyboard.

Your final project should look like the figure below:




| | Task Mode | Task Name | Duration | Predecessors | Successors | Start |
|---|---|---|---|---|---|---|
| 0 | | My New House | 204 days | | | Mon 10/5/15 |
| 1 | | Acquire land | 30 days | | 2 | Mon 10/5/15 |
| 2 | | Create architectural design | 20 days | 1 | 3 | Mon 11/16/15 |
| 3 | | Acquire loan | 10 days | 2 | 4 | Wed 12/16/15 |
| 4 | | Acquire permits | 38 days | 3 | 5 | Thu 1/7/16 |
| 5 | | Excavate land | 12 days | 4 | 6 | Tue 3/1/16 |
| 6 | | Pour foundation | 5 days | 5 | 7 | Thu 3/17/16 |
| 7 | | Install house frame | 15 days | 6 | 8,10,15 | Thu 3/24/16 |
| 8 | | Install cabling | 10 days | 7 | 9SS | Thu 4/14/16 |
| 9 | | Install plumbing | 14 days | 8SS | | Thu 4/14/16 |
| 10 | | Install HVAC | 8 days | 7 | 11 | Thu 4/14/16 |
| 11 | | Install roofing | 5 days | 10 | 12 | Tue 4/26/16 |
| 12 | | Installing siding | 15 days | 11 | 13 | Tue 5/3/16 |
| 13 | | Install drywall | 15 days | 12 | 14,16 | Tue 5/24/16 |
| 14 | | Install furnishings | 10 days | 13 | | Tue 6/14/16 |
| 15 | | Hire interior designer | 5 days | 7 | | Thu 4/14/16 |
| 16 | | Paint walls | 7 days | 13 | 17 | Tue 6/14/16 |

The final project with predecessors.

# 5. Coding and Testing

## 5.1 Implementation :-

### MODULE 1 :- SIGNUP.aspx.cs

### CODING :-

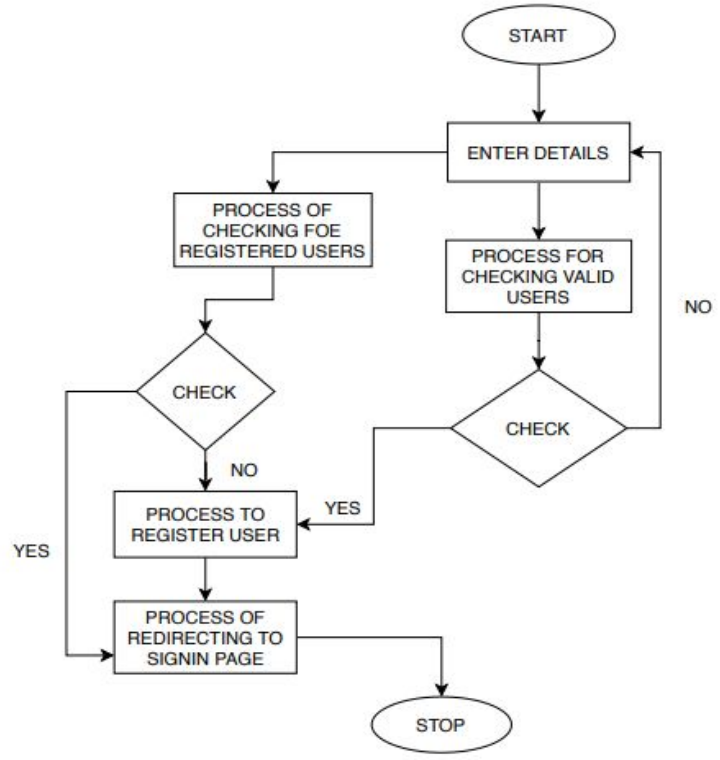
```
       public partial class WebForm2 : System.Web.UI.Page
        {
            SqlCommand cmd = new SqlCommand();
            SqlConnection con = new SqlConnection();
            SqlConnection con1 = new SqlConnection();
            protected void Page_Load(object sender, EventArgs e)
            {
                con.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=dips;integrated Security=true";
                con.Open();

            }

            protected void submit_Click(object sender, EventArgs e)
            {
                SqlCommand cmdv = new SqlCommand("select aadhar_num from users where
aadhar_num=" + txtaadhars.Text, con);
                SqlDataReader drav = cmdv.ExecuteReader();
                if (!drav.Read())                          //check if user exists
                {
                    con1.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=aadhar;integrated Security=true";
                    con1.Open();
                    drav.Close();
                    SqlCommand cmd1 = new SqlCommand("Select fullname from aadhar where
u_id=" + txtaadhars.Text, con1);
                    SqlDataReader dra = cmd1.ExecuteReader();
                    if (dra.Read())                                                  //check
if aadhar number correct
                    {
                        string name = dra.GetValue(0).ToString();
                        string entered_name = txtnames.Text.ToString();
                        if (string.Equals(entered_name, name))                       //check
if aadhar name matches with entered name
                        {
                            String pan = "0", voter = "0";
                            SqlCommand cmd = new SqlCommand("insert into users" +
"(fullname,mobile_num,aadhar_num,dob,pwd,pan_num,voter_num)values(@fullname,@mobile_num,@aadh
ar_num,@dob,@pwd,@pan_num,@voter_num)", con);
                            cmd.Parameters.AddWithValue("@fullname", txtnames.Text);
                            cmd.Parameters.AddWithValue("@mobile_num", txtnumbers.Text);
                            cmd.Parameters.AddWithValue("@aadhar_num", txtaadhars.Text);
                            cmd.Parameters.AddWithValue("@dob", txtdates.Text);
                            cmd.Parameters.AddWithValue("@pwd", txtpasswords.Text);
                            cmd.Parameters.AddWithValue("@pan_num", pan);
                            cmd.Parameters.AddWithValue("@voter_num", voter);
                            cmd.ExecuteNonQuery();
                            Page.ClientScript.RegisterStartupScript(this.GetType(),
"Scripts", "<script>alert('Successfully Registered Please proceed to sign
in');window.location='signin.aspx';</script>");
```

```
                        //Response.Write("<script>alert('Click Ok to SignIN')</script>");
                        //Response.Redirect("signin.aspx");
                    }
                }
                else
                {
                    Page.ClientScript.RegisterStartupScript(this.GetType(), "Scripts",
"<script>alert('Please enter correct aadhar number')</script>");
                    //Response.Redirect("index.aspx");
                }
            }
            else
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(), "Scripts",
"<script>alert('Already Registered Please proceed to sign
in');window.location='signin.aspx';</script>");
            }
        }
    }
```

## TESTING :-

**Cyclomatic Complexity :-** Cyclomatic complexity is a software metric used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code. It was developed by Thomas J. McCabe, Sr. in 1976.

## Cyclomatic Complexity [ Graph / flowchart ]:-




## Cyclomatic Complexity [ Calculation ]:-

Cyclomatic complexity = E - N + 2*P

where,

E = number of edges in the flow graph. = 10

N = number of nodes in the flow graph. = 9

P = number of nodes that have exit points = 1

**Therefore Ans :- 10 – 9 + 2(1) = 3**

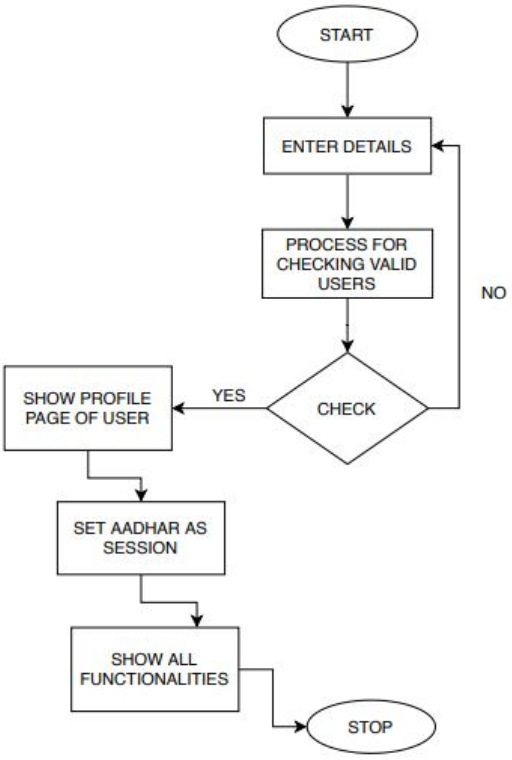
## MODULE 2 :- SIGNIN.aspx.cs

## CODING :-

```
public partial class WebForm3 : System.Web.UI.Page
    {
        SqlCommand cmd = new SqlCommand();
        SqlConnection con = new SqlConnection();
        protected void Page_Load(object sender, EventArgs e)
        {
            con.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=dips;integrated Security=true";
            con.Open();
        }

        protected void submit_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("select * from users where aadhar_num='"+
txtaadhar.Text + "' and pwd='"+ txtpassword.Text +"'", con);
            SqlDataReader dr = cmd.ExecuteReader();
            if (dr.Read())
            {
                Session["id"] = txtaadhar.Text;
                Response.Redirect("dashboard.aspx");
                //con.Close();
            }
            else
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(), "Scripts",
"<script>alert('Invalid Aadhar-Number or
Password');window.location='signin.aspx';</script>");

            }
        }
           }
```

TESTING :-

## Cyclomatic Complexity [ Graph / flowchart ]:-




## Cyclomatic Complexity [ Calculation ]:-

Cyclomatic complexity = E - N + 2*P

where,

E = number of edges in the flow graph. = 8

N = number of nodes in the flow graph. = 8

P = number of nodes that have exit points = 1

**Therefore Ans :- 8 – 8 + 2(1) = 2**

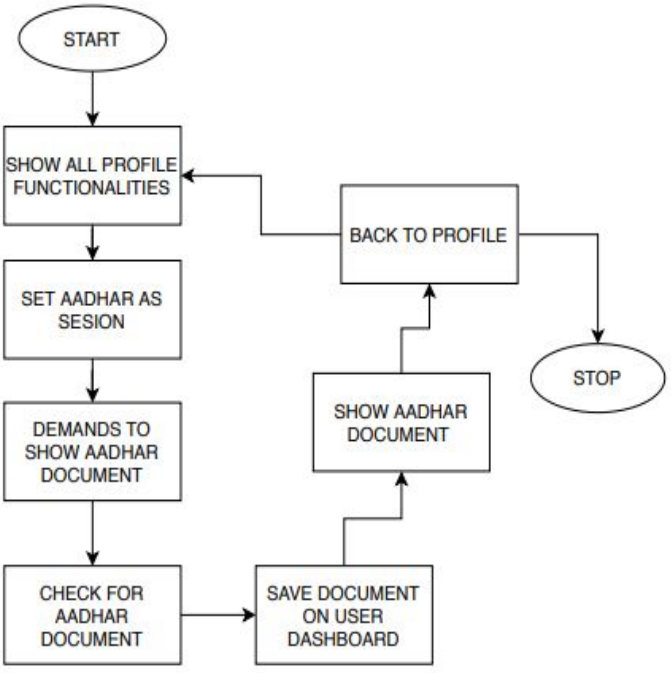
## MODULE 3 :- AADHAR.aspx.cs

## CODING :-

```
public partial class WebForm4 : System.Web.UI.Page
    {

        protected void Page_Load(object sender, EventArgs e)
        {
            //get aadhar number and name from USER database to display information
            var sb = new System.Text.StringBuilder();
            String id;
            id = Session["id"] as String;

            //Int64 uid=Int64.Parse(id);

            sb.Append("<br/><br/><br/><br/><br/><br/>");
            Response.Write("AADHAR id:" + id);
            Response.Write("AADHAR id:" + Session["id"]);

            //string co = ConfigurationManager.ConnectionStrings["aadhar"].ConnectionString;

            SqlConnection cona = new SqlConnection();
            cona.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=aadhar;integrated Security=true";
            cona.Open();
            SqlCommand cmda = new SqlCommand("select * from aadhar where u_id='" + id + "'",
cona);

            //SqlDataReader dra = cmda.ExecuteReader();

            SqlDataAdapter sda = new SqlDataAdapter(cmda);

            DataTable dt = new DataTable();
            sda.Fill(dt);
            GridView1.DataSource = dt;
            GridView1.DataBind();
            cona.Close();


        }

        protected void Back_Click(object sender, EventArgs e)
        {
            Response.Redirect("dashboard.aspx");
        }

        protected void pan_Click(object sender, EventArgs e)
        {
            Response.Redirect("panCheck.aspx");
        }

        protected void voter_Click(object sender, EventArgs e)
        {
            Response.Redirect("voterCheck.aspx");
        }
    }
```

TESTING :-

## Cyclomatic Complexity [ Graph / flowchart ]:-




## Cyclomatic Complexity [ Calculation ]:-

Cyclomatic complexity = E - N + 2*P

where,

E = number of edges in the flow graph. = 9

N = number of nodes in the flow graph. = 9

P = number of nodes that have exit points = 1

**Therefore Ans :- 9 – 9 + 2(1) = 2**

## MODULE 4 :- PANCHECK.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
        {
            //get aadhar number and name from USER database to display information
            var sb = new System.Text.StringBuilder();
            String id;
            id = Session["id"] as String;

            //Int64 uid=Int64.Parse(id);

            sb.Append("<br/><br/><br/><br/><br/><br/>");
            Response.Write("AADHAR id:" + id);
            Response.Write("AADHAR id:" + Session["id"]);

            //string co = ConfigurationManager.ConnectionStrings["aadhar"].ConnectionString;

            SqlConnection con = new SqlConnection();
            con.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=dips;integrated Security=true";
            con.Open();
            SqlCommand cmd = new SqlCommand("select * from users where aadhar_num='" + id +
"'", con);

            SqlDataReader dra = cmd.ExecuteReader();
            dra.Read();
            string pan = dra.GetValue(5).ToString();
            string panz = "0";
            if (pan.Equals(panz))
            {
                //update pan set images = PAN / DHRUVILSHAH.jpg where aadhar_num =
975697580296;
                Response.Redirect("pantext.aspx");
                dra.Close();
            }
            else
            {
                Response.Redirect("panimage.aspx");
                dra.Close();
            }



            con.Close();


        }
    }
```

## PANTEXT.aspx.cs :-

```
public partial class WebForm7 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void submit_Click(object sender, EventArgs e)
        {
```

```
            var sb = new System.Text.StringBuilder();
            String id;
            id = Session["id"] as String;

            //Int64 uid=Int64.Parse(id);

            sb.Append("<br/><br/><br/><br/><br/><br/>");
            Response.Write("AADHAR id:" + id);
            Response.Write("AADHAR id:" + Session["id"]);

            Session["pid"] = txtpan.Text;
            string pid;
            pid = Session["pid"] as String;

            SqlConnection conp = new SqlConnection();
            conp.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=aadhar;integrated Security=true";
            conp.Open();
            SqlCommand cmdp = new SqlCommand("select * from pan where pan_num='" + pid + "'",
conp);
            SqlDataReader dr = cmdp.ExecuteReader();
            if (dr.Read())
            {
                string panid = dr.GetValue(1).ToString();

                if (pid.Equals(panid))
                {
                    SqlConnection con = new SqlConnection();
                    con.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=dips;integrated Security=true";
                    con.Open();
                    SqlCommand cmd = new SqlCommand("UPDATE users SET pan_num = '" + pid + "'
WHERE aadhar_num = '" + id + "'", con);
                    cmd.ExecuteNonQuery();
                    Response.Redirect("dashboard.aspx");
                }
            }
        }
    }
```

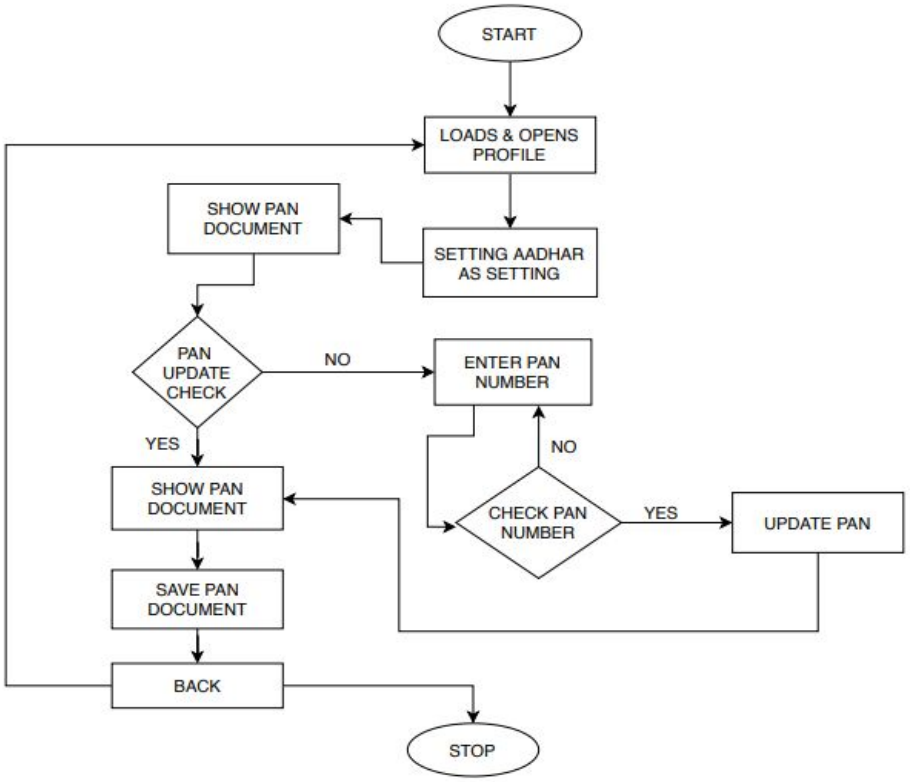
PANIMAGE.aspx.cs :-

```
       public partial class WebForm8 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            var sb = new System.Text.StringBuilder();
            String id;
            id = Session["id"] as String;

            //Int64 uid=Int64.Parse(id);

            sb.Append("<br/><br/><br/><br/><br/><br/>");
            Response.Write("AADHAR id:" + id);
            Response.Write("AADHAR id:" + Session["id"]);
            SqlConnection conp = new SqlConnection();
            conp.ConnectionString = "Data source=DESKTOP-CAN344L;initial
catalog=aadhar;integrated Security=true";
            conp.Open();
```

```
            SqlCommand cmdp = new SqlCommand("select * from pan where aadhar_num='" + id +
"'", conp);
            SqlDataAdapter sda = new SqlDataAdapter(cmdp);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            GridView1.DataSource = dt;
            GridView1.DataBind();
        }

        protected void aadhar_Click(object sender, EventArgs e)
        {
            Response.Redirect("aadhar.aspx");
        }

        protected void back_Click(object sender, EventArgs e)
        {
            Response.Redirect("dashboard.aspx");
        }

        protected void voter_Click(object sender, EventArgs e)
        {
            Response.Redirect("votercheck.aspx");
        }
    }
```

TESTING :-

## Cyclomatic Complexity [ Graph / flowchart ]:-




## Cyclomatic Complexity [ Calculation ]:-

Cyclomatic complexity = E - N + 2*P

where,

E = number of edges in the flow graph. = 14

N = number of nodes in the flow graph. = 12

P = number of nodes that have exit points = 1

**Therefore Ans :- 14 – 12 + 2(1) = 4**

## 5.2 Examine various code analysis tools and prepare a comparative analysis.

Back in the day, we'd write some code, compile, execute, see what happened and repeat. That was testing. (Sometimes that's still what testing looks like, for better or worse.) Today, we can do a lot better.

Today we have IntelliSense, linters, real-time code analysis, automated test builders, automated test runners, test coverage analysis.... The list goes on, and that's just the tools we use to make sure code is as bulletproof as possible before we run a build. From build and compile we have debugging tools and more. These days there is very little reason for your code to include architectural or language-level errors.

Visual Studio has always included debugging tools and added built-in code analysis and testing tools since at least Visual Studio 2012. These tools just keep getting better with each new release, but they don't cover every use case, and some don't get updated as frequently as we'd like to cover new language features or code architectures.

### Linters and Code Analysis

**GCop** is a fairly new set of C# code analysis rules (with really nice **setup, use and rules documentation**) from **Geeks Ltd.**, which may be worth checking out if you're not entirely satisfied with other code analysis rulesets (or perhaps using alongside of those other rules for extended coverage). GCop is intended to be installed in your project as a NuGet package. To allow for rules that can't run from a package, Paymon (a co-founder of Geeks Ltd.) has released a **GCop.Extra** Visual Studio 2017 extension for use with GCop. The extension enables GCop rules such as Minimum Scope that detect whether methods that are more visible than they need to be (private versus internal versus public).

**MultiLinter**, enables you to replace the (already outdated) linters built into Visual Studio 2017 with the standard linters available through Node.js including (but not limited to) ESLint, JSLint, JSHint, Stylelint, CssLint and Sass-lint. MultiLinter lets you turn verbose debugging on, configure which linters to use (including running multiple linters against a file at the same time), update linters and linting rules, set rule severity warnings, and much more.

**XamRight**, is an extension for Visual Studio 2015 and 2017 that brings design-time code analysis and coding assistance to Xamarin.Forms XAML development. You get IntelliSense, warnings, view model and data binding analysis and debugging, navigation tools for moving between XAML and C# model definitions, custom view implementations and more, along with navigation from XAML resource references to definitions. XamRight can analyze your own model-view model binding, but also includes built-in support for popular MVVM frameworks including MVVMCross, MVVMLight, FreshMVVM, Prism and Caliburn.Micro. A 30-day free trial is available and licensing is available on a monthly or yearly basis.

**NDepend** (see **Figure 1**), one of the most popular commercial static code analysis tools for .NET Framework development, recently released a substantial update including support for .NET Core 2.1, ubiquitous language checks in Domain Driven Design (DDD), performance improvements for Visual Studio 2017 and over a dozen new or improved code analysis rules. A key new feature for NDepend is real-time technical debt estimation that's updated as you code. Proud of that new method? Guess what, you just added 30 minutes of future technical debt. Maybe check NDepend's analysis and spend a minute refactoring. I love it. NDepend offers a free 14-day trial and per-developer or per-build machine licensing.

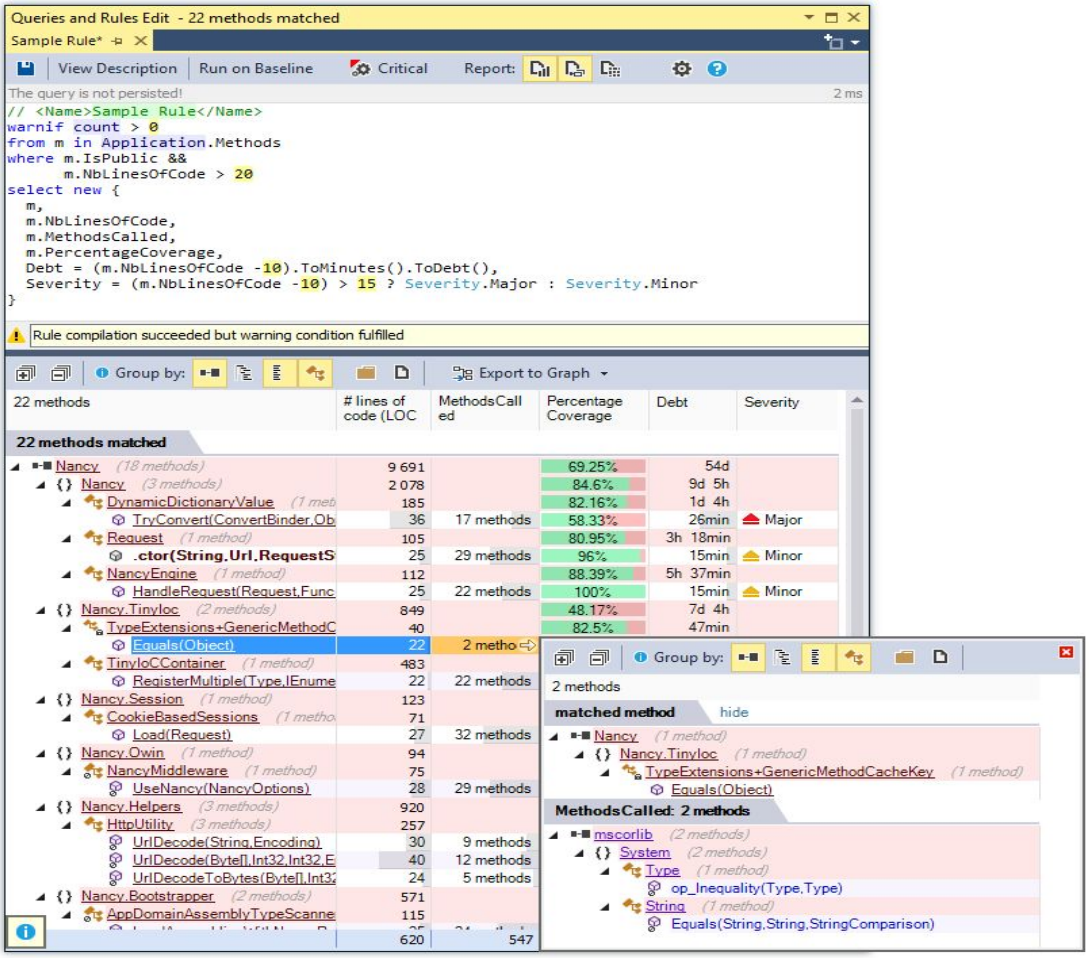



***Figure 1.*** NDepend's Static Analysis Tools Have Been Updated for .NET Core 2

**Async Method Name Fixer** is an effective little tool for doing a simple but often overlooked task: making sure your async methods are named appropriately. In a nutshell, it looks for methods defined as async and, if you haven't given the method a name with "Async" on the end, the extension flags all instances of the method and calls to it.

## Code Coverage and Testing

**AxoCover**, provides Visual Studio integration for code coverage and unit testing with **OpenCover**. AxoCover lets you run, debug and check code coverage for unit tests in .NET Framework projects for Windows. You can browse and analyze coverage by test in a hierarchical view and dig down into line-by-line coverage and test results. AxoCover supports the MSTest, xUnit and NUnit test frameworks.

**SmartTests.Extension**, is an extension to show current and missing tests for NUnit, Xunit and MSTest within Visual Studio. SmartTests.Extension integrates into your Visual Studio environment the **Pretty Objects** SmartTests library, which helps you write smart unit tests, and the SmartTests.Analyzer, a Roslyn Analyzer to display missing tests as warnings. The extension shows tests in a centralized window, lets you see current tests and any missing tests, and lets you navigate quickly to specific tests.

SmartBear's **TestLeft** UI test automation framework (see **Figure 2**) has been updated to support Visual Studio 2013, 2015 and 2017. TestLeft integrates directly into the Visual Studio development environment, enabling you to create tests as you code. A built-in object spy gives you the ability to create tests that support over 500 common Web and desktop UI controls. You can create automated tests for a variety of popular development frameworks including .NET Framework, Winforms, WPF, Java, HTML5 and AngularJS. Further test coverage includes controls from Infragistics, DevExpress, Syncfusion and Telerik, along with cross-browser testing with legacy versions of Internet Explorer, Edge, Firefox and Chrome. Tests created in TestLeft can be migrated into **TestComplete** for automated testing as part of your continuous integration and release management processes. Register online for a free 30-day trial. Yearly per-node and floating licenses are available from SmartBear.

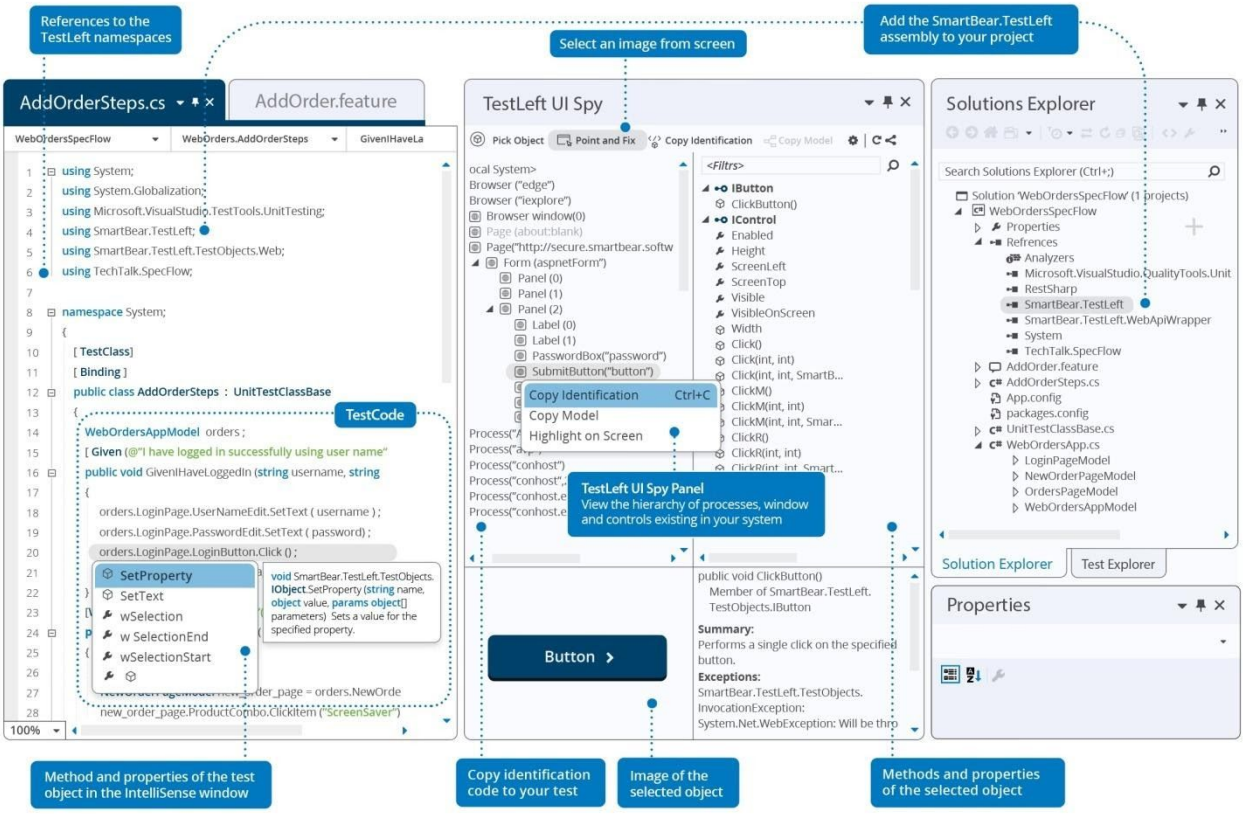



***Figure 2.***TestLeft Simplifies the Creation of UI Test Automation as You Code

## Better Debugging

**VisualSOS.Extension** (see **Figure 3**) gives you access to features of the **Microsoft SOS Debugging Extension** and Windbg that are not available directly from the Visual Studio Debugger. VisualSOS.Extension also gives you menu access to those features instead of having to remember the commands and option flags. Visual SOS is available as both a Visual Studio 2017 extension and a stand-alone debugger. To learn more, see Hernandez's blog post **Visual SOS – Visual Studio extension to debug managed applications through SOS** for an overview and some tips for more effective debugging with SOS.

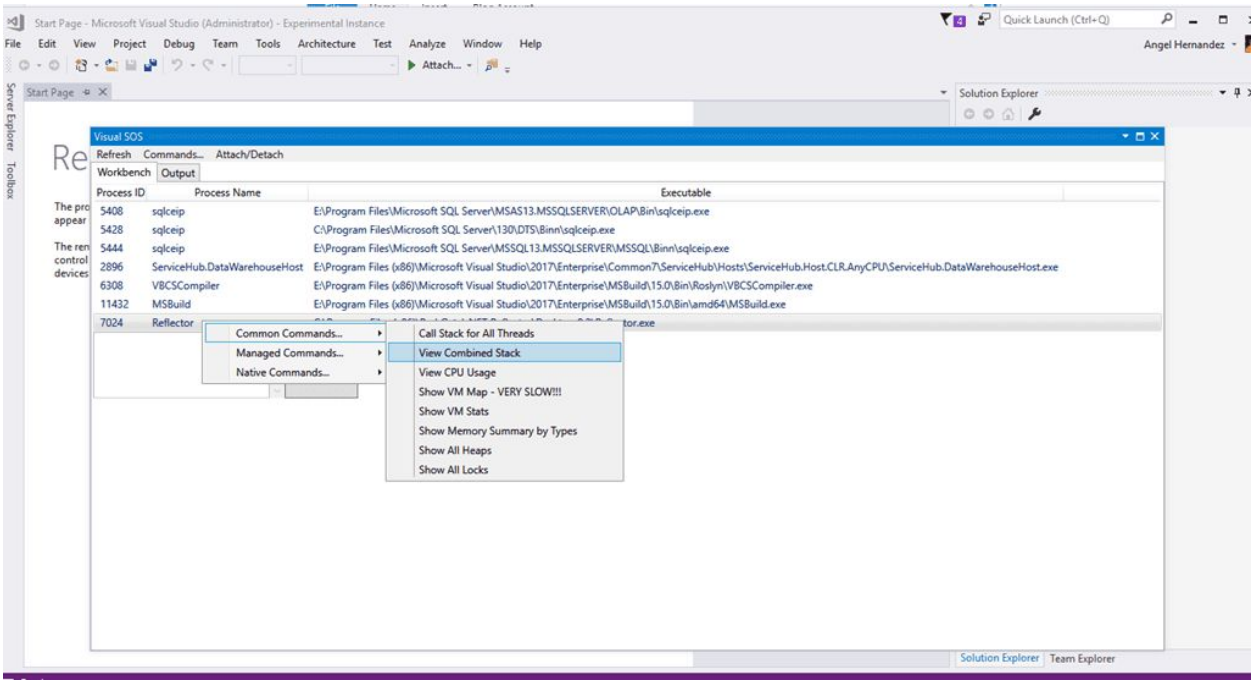



***Figure 3.*** VisualSOS Adds SOS and WinDbg Debugging Tools to Visual Studio

**LINQBridgeVs**, from Coding Adventures, provides a Custom Debugger Visualizer within Visual Studio that creates a bridge between your Visual Studio debugging session and the external **LINQPad** scratchpad and test environment. After rebuilding your project, you can right-click on any public classes or structs in the project and open the related debugging data within LINQPad. The extension supports Visual Studio 2012, 2013, 2015 and 2017 along with LINQPad 4 or 5, but is limited to .NET Framework 4.0 and above. Check the documentation for installation and troubleshooting notes.

Microsoft's **VS Live Share**, currently in preview (see **Figure 4**), provides a collaborative development environment for Visual Studio 2017, enabling you to share code, collaboratively edit, securely share local servers and even collaboratively debug your code in real-time. It's not a screen

share service or centralized codebase; you're able to work independently in your local Visual Studio environment while collaborating on code editing and debugging. As we go to press, the VS Live Share is in limited preview: anyone can download the extension, run it and join a session, but permission to share a session requires registration and acceptance into the preview. For more details about VS Live Share including demos, see the **Visual Studio Live Share** Web site and **Visual Studio Live Share Microsoft Docs**.

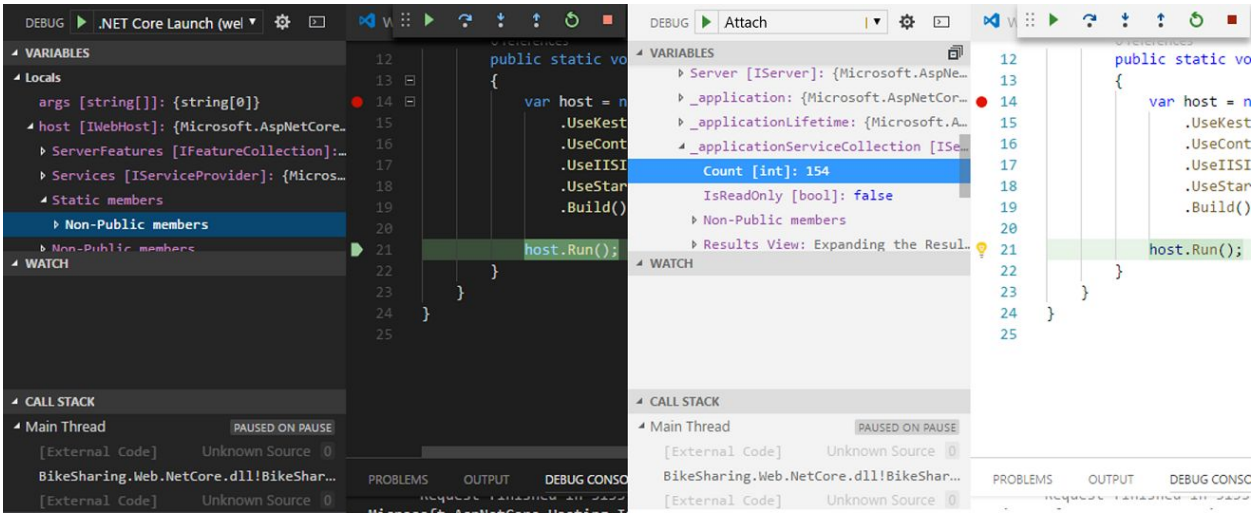



***Figure 4.***VS Live Share Lets You Join Collaborative Editing and Debugging Sessions

**Inlining Analyzer**, by Stephan Zehetner, helps analyze and streamline debugging for methods inlined by the JIT compiler. Inlining can be a powerful performance enhancement for your code, but getting the best results requires careful coding, testing and time-consuming performance testing. Inlining Analyzer simplifies the task by scanning your code, highlighting methods that will be inlined, providing warnings and tooltips for methods that fail to inline, and streamlining the JIT compilation tasks needed to test for inline method tracing.

## More Related Tools and Extensions

**StyleCop Check-in Policy** has been updated with support for Visual Studio 2012 through 2017. It's a simple extension that lets you toggle StyleCop warnings so they don't get in the way for incremental local builds, but can be turned back on to fix issues before checking in your code.

**Sharpen** is a new Visual Studio 2017 extension that aims to help you introduce new C# features into your code in an intelligent manner. Sharpen analyzes your code and determines where new C# features might be useful, but it doesn't just try to jam new features into your project. You'll see both the pros and cons of adding the feature before committing. If you want to use the feature, Sharping provides one-click refactoring to use the feature consistently throughout your project. Sharpen is open source and free!

There's a lot of code analysis data being created by all these tools. What if you could harness it for further meta-analysis, visualization, or other purposes? An interesting step in this direction is demonstrated by **Microsoft SARIF Viewer** for Visual Studio 2015 and 2017, by Microsoft DevLabs. The extension gives you a UI for viewing static analysis log files created using the new **Static Analysis Results Interchange Format (SARIF)** standard. The SARIF Viewer extension displays analysis results in the Visual Studio Error List and result details in a dockable tool window.

You can right-click a highlighted error line to open the target file. If you find this interesting, take a closer look at the **SARIF .NET SDK** and **SARIF specification** to get an idea what my be possible for code analysis logging, data meta-analysis and data visualization, not to mention the possibility of componentized code analysis tooling.

While not directly related to testing and debugging, **Dotfuscator**, from PreEmptive Solutions, is an important tool for safely releasing your code into the world. Dotfuscator is a code obfuscater and application hardener specifically tailored to .NET Framework-based projects including Xamarin, Unity, ASP.NET and UWP apps. In addition to renaming, string encryption, metadata stripping and other transformations, Dotfuscator also compacts your code to reduce application size, watermarks your application with copyright information to combat intellectual property theft, and can even detect unauthorized tampering and debugging. A free trial and licensing information can be found on the **PreEmptive Solutions** Web site.

### A Note To Extension Developers

A tip for extension developers from a developer who uses them and writes about them: If you want developers to use your tools, it's really, really important to provide documentation. At the very least explain what your tool is supposed to accomplish and its basic requirements and dependencies. Optimally you'll explain what it does, why it's useful, an overview of features, how to use it and how to troubleshoot problems.

Whether you're sharing your tools, libraries, or extensions on Visual Studio Marketplace, GitHub, or your own Web site, it's absolutely crucial to your users to provide solid documentation. Also, I simply ignore tools that are undocumented. You won't see them here no matter how potentially useful or revolutionary.

5.3 Examine various Testing tools (Win runner, Load runner) and prepare a comparative analysis.

Win Runner

Win Runner software is an enterprise automation functional GUI testing tool from Mercury Interactive Corporation that allows a user to record and play back UI interactions as test scripts. It is the most commonly used automated testing tool.

Mercury (formerly Mercury Interactive Corporation) is now part of Hewlett-Packard.

Win Runner was originally written by Mercury Interactive. Mercury Interactive was subsequently acquired by Hewlett Packard (HP).

HP acquired Load Runner as part of its acquisition of Mercury Interactive in November 2006. Hence, now it is known as HP Win Runner.

As a functional test suite, it worked with HP QuickTest Professional and supported enterprise quality assurance.

It captured, verified and replayed user interactions automatically, in order to identify defects and determine whether business processes worked as designed.

The software implemented a proprietary Test Script Language (TSL) that allowed customization and parameterization of user input.

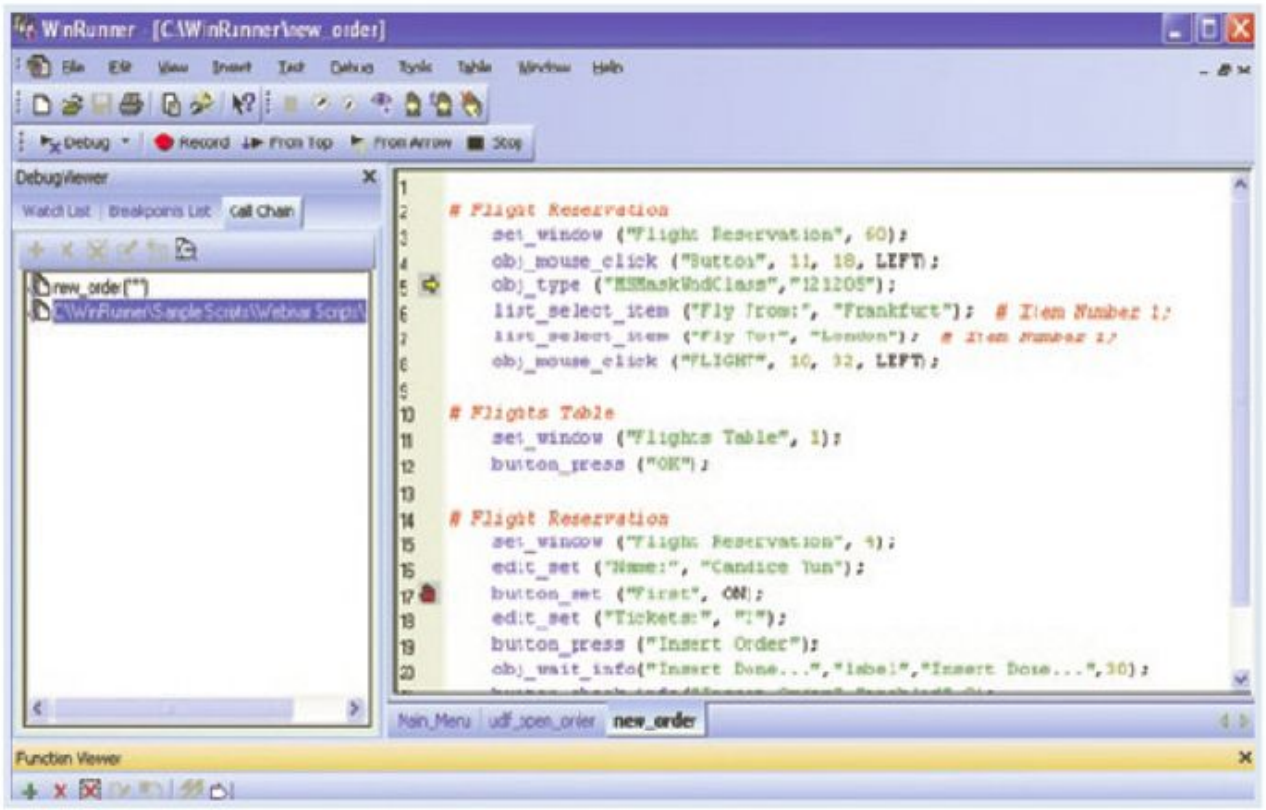
Working with Win-Runner

- Win Runner must learn to recognize the objects in your application in order to run tests.
- Win Runner writes scripts automatically when you record actions on your application, or you can program directly in Mercury Interactive's Test Script Language (TSL).

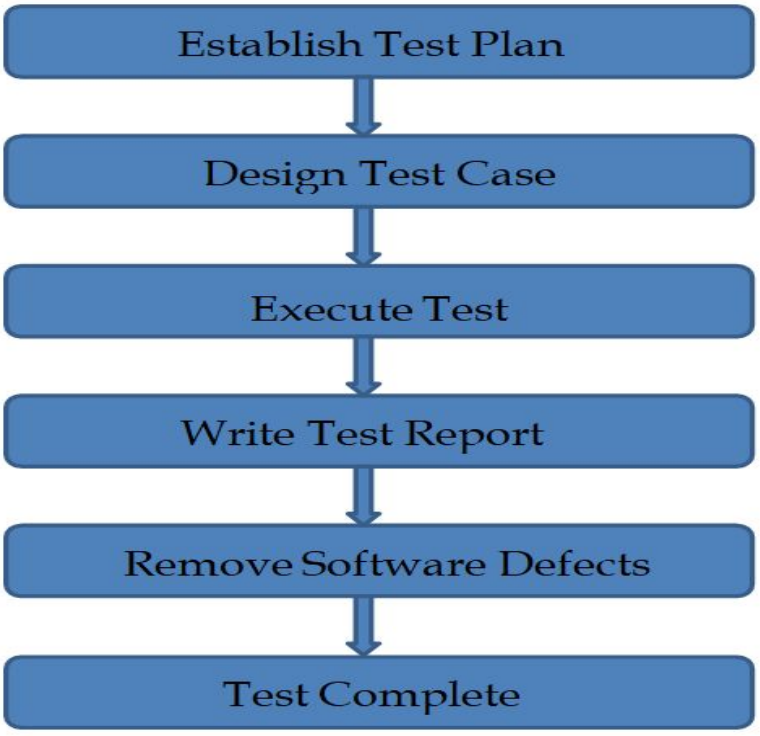
- You debug the tests to check that they operate smoothly and without interruption.
- You run the tests on a new version of the application in order to check the application's behavior.
- You examine the test results to pinpoint defects in the application.

Features of Win-Runner

- Automatic Recovery: It is possible for the user to set up various operations that will be activated if an exception event appears. The recovery manager will give you a wizard that will allow you to setup a scenario for recovery.
- Support for Various Environments: It includes support for Internet Explorer 6.x and Netscape 6.x, Windows XP and Sybase's Power Builder 8 in addition to 30+ environments.
- Cost Effective: It provides the most powerful, productive and cost-effective solution for verifying enterprise application functionality.
- Support Multiple Data Combination: It has an ability to use numerous data combinations for one test. The DataDriver Wizard has been designed for automatic processing of large amounts of data.
- The default time setting of Win runner is 10000 ms.
- The logical name of true and false in Win runner is 1 & 0.
- Win Runner doesn't provide the snapshot of output.
- Win Runner has 3 kinds of Checkpoints,2 kinds of Recordings, and 3 kinds of Exception Handling mechanisms.
- In Win Runner GUI Spy is available.




Process-Flow of Win-Runner



Load Runner

LoadRunner is a software testing tool from Micro Focus. It is used to test applications, measuring system behavior and performance under load. LoadRunner can simulate thousands of users concurrently using application software, recording and later analyzing the performance of key components of the application.

LoadRunner simulates user activity by generating messages between application components or by simulating interactions with the user interface such as keypresses or mouse movements. The messages and interactions to be generated are stored in scripts. LoadRunner can generate the scripts by recording them, such as logging HTTP requests between a client web browser and an application's web server.

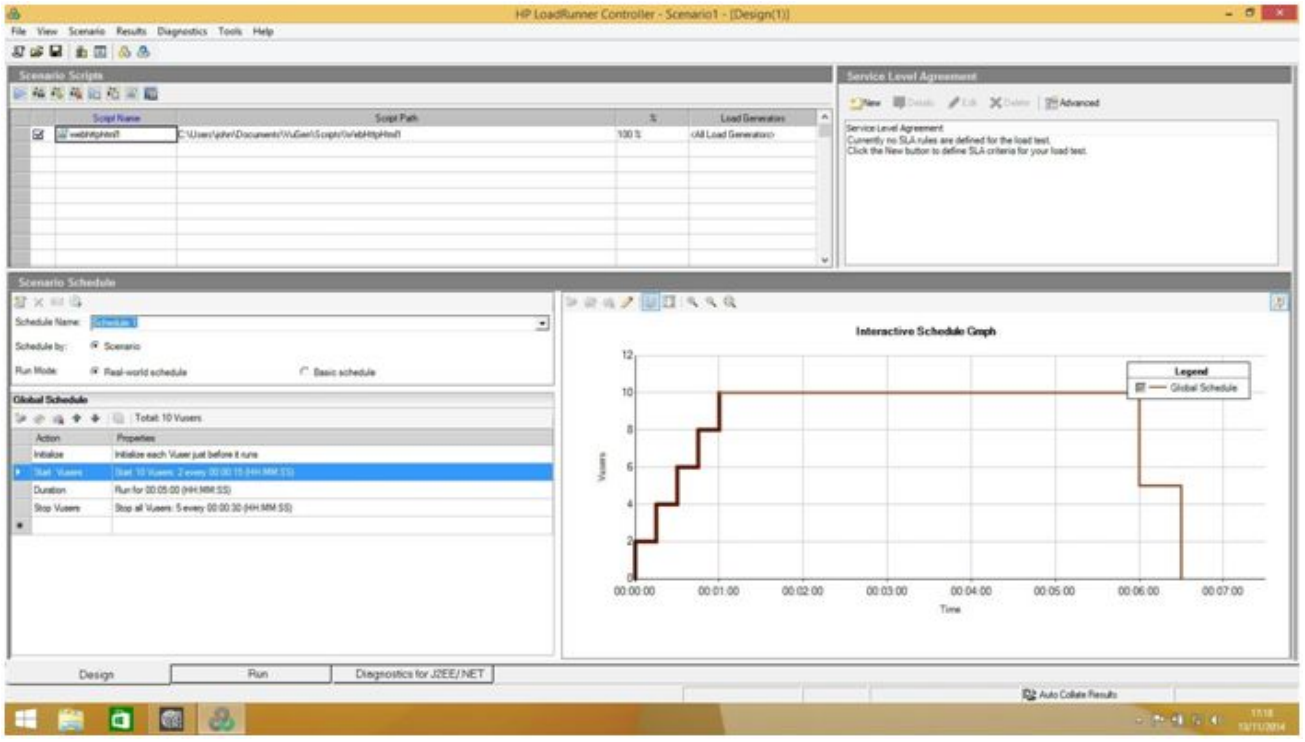
Hewlett Packard Enterprise acquired LoadRunner as part of its acquisition of Mercury Interactive in November 2006. In Sept 2016, Hewlett Packard Enterprise announced it is selling its software business, including Mercury products, to Micro Focus. As of 01-Sept-2017, the acquisition was complete.

Working with Load-Runner

- The Virtual User Generator captures end-user business processes and creates an automated performance testing script, also known as a virtual user script.
- The Controller organizes, drives, manages, and monitors the load test.
- The Load Generators create the load by running virtual users.
- The Analysis helps you view, dissect, and compare the performance results.
- The Launcher provides a single point of access for all of the Load Runner components.

Features of Load-Runner

- It has excellent monitoring and analysis interface where tester can see reports in easy to understand colored charts and graphics.
- It uses C as default programming language. However, it also supports other languages like Java and Visual Basic.
- No need to install it on the server under test. It uses native monitors.
- It has a support for most of the commonly used protocols.
- It has GUI generated scripts which can be modified as per the requirements.
- The tool can quickly point out the effect of the wide area network (WAN) on application reliability, performance and response time.
- Supports most of the protocols
- Excellent tutorial, exhaustive documentation and active tool support from HP.



Process-Flow of Load-Runner