SOFTWARE ENGINEERING

ONLINE FOOD DELIVERY SYSTEM

PROJECT PLANNING

INTRODUCTON:

Our PHP-based Online Food Delivery System enables both external and internal clients (admins or employees) to interact with each other. Admin users can receive orders from users of our website, and users can choose what they would like to order from a large variety of food businesses. They may add multiple items from the same restraint and have it delivered to their doorstep. The option of adding more than one address (in the event that the user has a home and work address) is also available. Multiple payment options such as UPI, cash on Delivery have been implemented to make our website efficient and user friendly.

Software-Hardware Integration: While there are many online food delivery systems on the market, it should be well integrated with hardware as well since not everyone may have access to smartphones. Due to this our website is laptop or pc compatible. Contactless booking and access to food is the main objective of this project. After the recent pandemic, many wish to order food from their favourite restraunts and eat it in the comfort of their homes. Some people may want to cancel their order, which we have made possible 3 minutes within placing the order. We also provide a feedback page, and a list of reasons for cancelling of the order. This helps us to improve backend glitches or problems if any. The aim is to automate its existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same.

1. PROCESS MODEL:

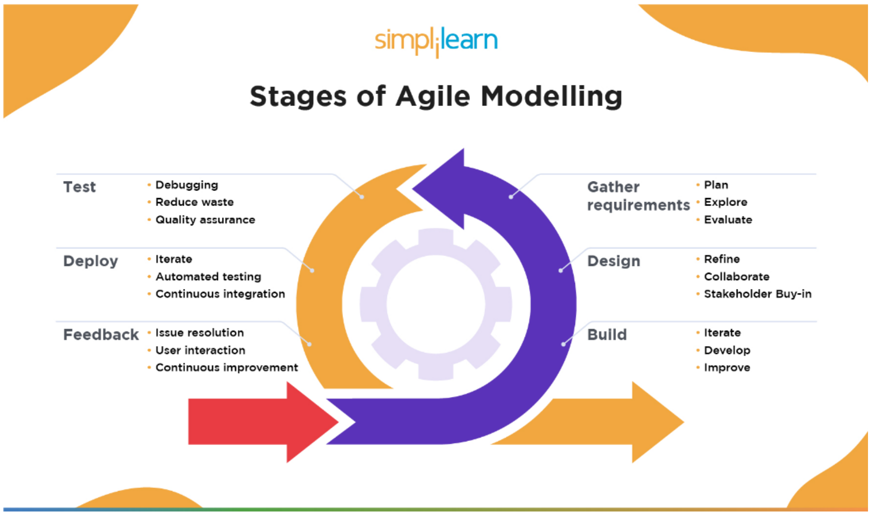
Our model is Agile Process model. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Following are the phases in the Agile model are as follows:

- ✓ Requirements gathering
- ✓ Design the requirements
- ✓ Construction/ iteration
- ✓ Testing/ Quality assurance

- ✓ Deployment
- ✓ Feedback



Stages of Agile Modelling

Test
- Debugging
- Reduce waste
- Quality assurance

Deploy
- Iterate
- Automated testing
- Continuous integration

Feedback
- Issue resolution
- User interaction
- Continuous improvement

Gather requirements
- Plan
- Explore
- Evaluate

Design
- Refine
- Collaborate
- Stakeholder Buy-in

Build
- Iterate
- Develop
- Improve

2. TOOLS USED:

Planning tool: Jira
Design tool: Powerpoint, Adobe
Version Control: GIT
Development tool: VS code, Git hub tools
Bug tracking: Jira Testing tool: Jira

3. DELIVERABLES:

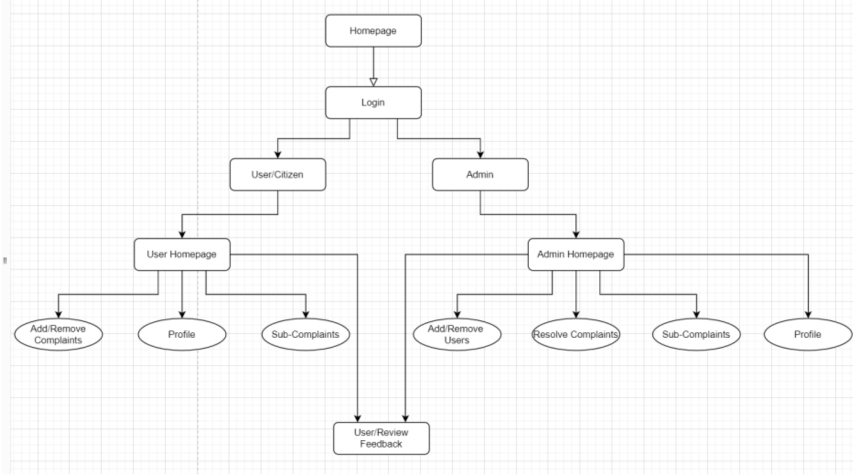
A website encompassing:

- ✓ A login page for user with otp 2- factor authentication
- ✓ A login page for admin with admin credentials
- ✓ A checkout page where user can review and place their order
- ✓ Payment page where mode of payment is chosen and payment is made

Code for the same

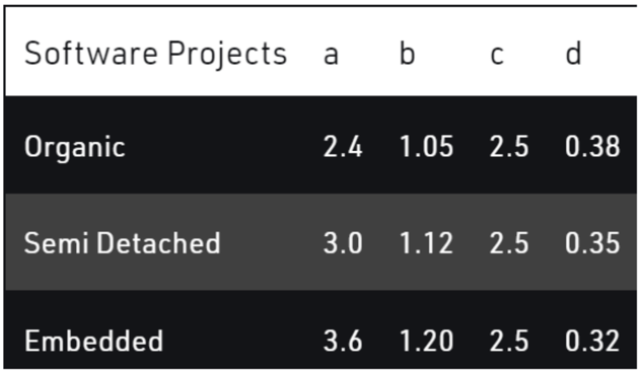
- ✓ User analytical tools integrated with the database

4. WORK BREAK DOWN STRUCTURE:



5. ESTIMATE OF EFFORT:

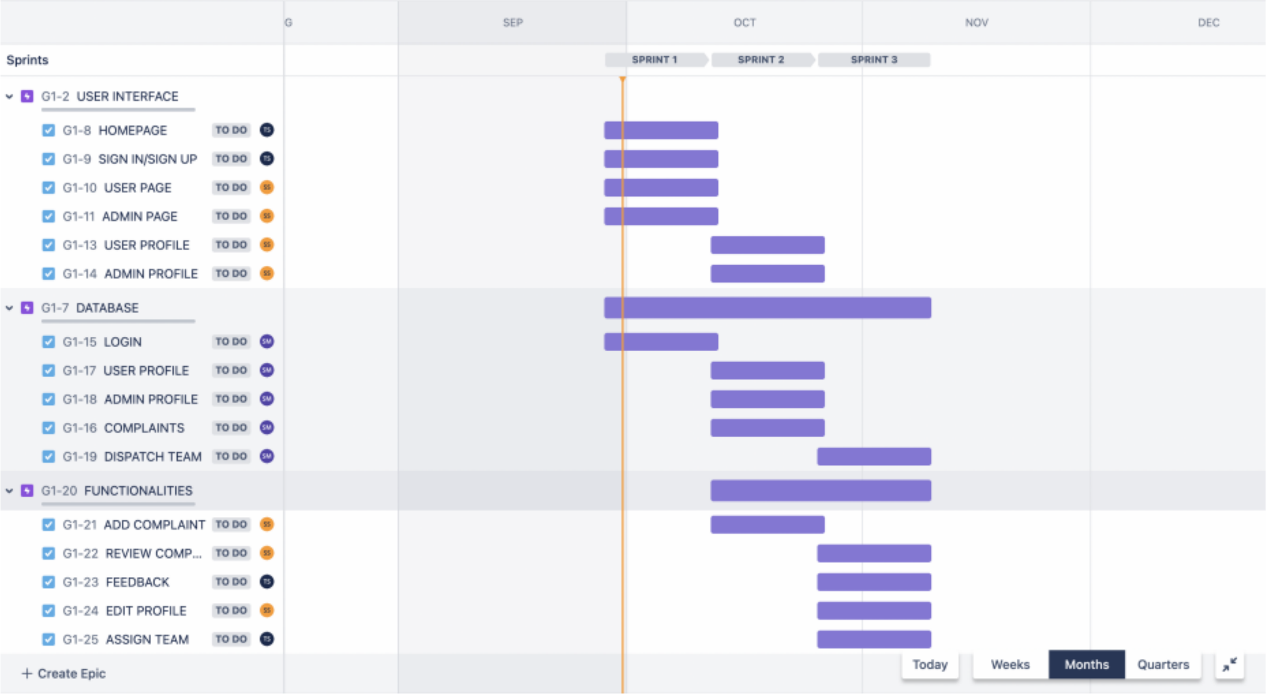
The problem statement of our project is well understood by us and it has been solved many times in the industry. The team members possess a decent knowledge in the domain. Our project would be classified as Organic.



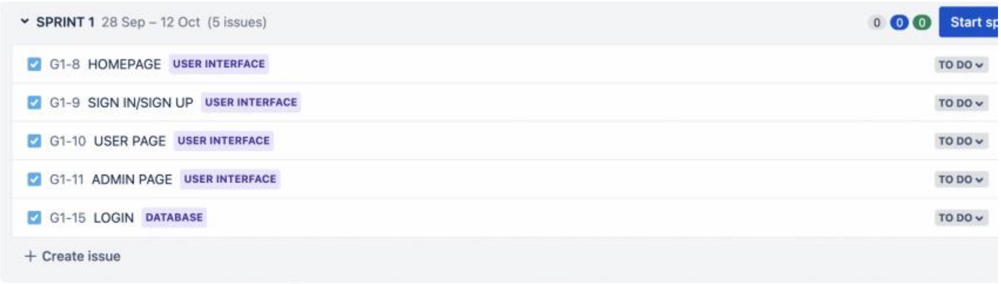
Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

KLOC = 2
A = 2.4, B = 1.05
Effort = a(KLOC)^b = 2.4*(2)^1.05 **= 4.96**

6. GANT CHART:


G
SEP
OCT
NOV
DEC
Sprints
SPRINT 1
SPRINT 2
SPRINT 3
G1-2 USER INTERFACE
G1-8 HOMEPAGE TO DO TS
G1-9 SIGN IN/SIGN UP TO DO TS
G1-10 USER PAGE TO DO SS
G1-11 ADMIN PAGE TO DO SS
G1-13 USER PROFILE TO DO SS
G1-14 ADMIN PROFILE TO DO SS
G1-7 DATABASE
G1-15 LOGIN TO DO SM
G1-17 USER PROFILE TO DO SM
G1-18 ADMIN PROFILE TO DO SM
G1-16 COMPLAINTS TO DO SM
G1-19 DISPATCH TEAM TO DO SM
G1-20 FUNCTIONALITIES
G1-21 ADD COMPLAINT TO DO SS
G1-22 REVIEW COMP... TO DO SS
G1-23 FEEDBACK TO DO TS
G1-24 EDIT PROFILE TO DO SS
G1-25 ASSIGN TEAM TO DO TS
+ Create Epic
Today
Weeks
Months
Quarters

7. PRODUCT BACKLOG:



SPRINT 1 28 Sep – 12 Oct (5 issues)

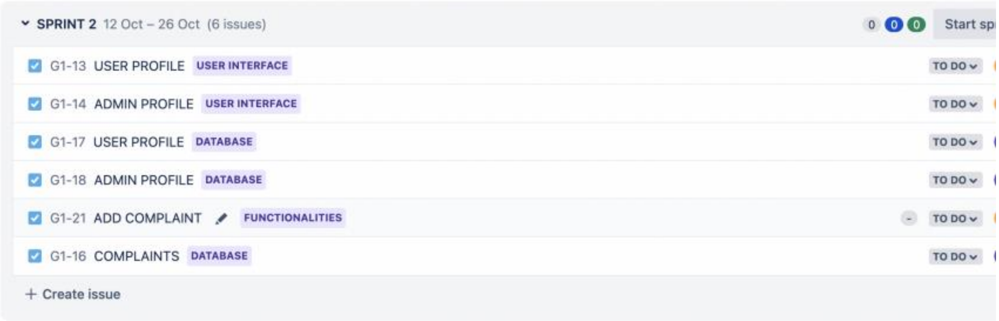
☑ G1-8 HOMEPAGE USER INTERFACE TO DO

☑ G1-9 SIGN IN/SIGN UP USER INTERFACE TO DO

☑ G1-10 USER PAGE USER INTERFACE TO DO

☑ G1-11 ADMIN PAGE USER INTERFACE TO DO

☑ G1-15 LOGIN DATABASE TO DO



SPRINT 2 12 Oct – 26 Oct (6 issues)

☑ G1-13 USER PROFILE USER INTERFACE TO DO

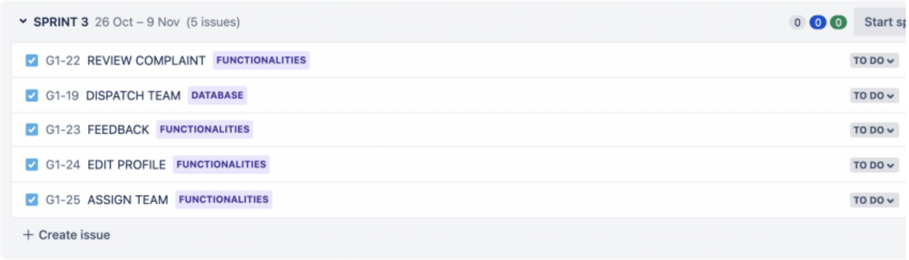
☑ G1-14 ADMIN PROFILE USER INTERFACE TO DO

☑ G1-17 USER PROFILE DATABASE TO DO

☑ G1-18 ADMIN PROFILE DATABASE TO DO

☑ G1-21 ADD COMPLAINT FUNCTIONALITIES - TO DO

☑ G1-16 COMPLAINTS DATABASE TO DO



SPRINT 3 26 Oct – 9 Nov (5 issues)

☑ G1-22 REVIEW COMPLAINT FUNCTIONALITIES TO DO

☑ G1-19 DISPATCH TEAM DATABASE TO DO

☑ G1-23 FEEDBACK FUNCTIONALITIES TO DO

☑ G1-24 EDIT PROFILE FUNCTIONALITIES TO DO

☑ G1-25 ASSIGN TEAM FUNCTIONALITIES TO DO