BRAINSTREAM SOFTWARE INC.

CS 350 SOFTWARE ENGINEERING I

FALL 2017 PROJECT

ALI KALKANDELEN

SAMUEL OWUSU-BINEY

STEPHEN KILNISAN

BrainSteam Intelligence Software Co.

In 2018, there will be a mid-term Congressional election for possible control of the House and Senate by the Democrats. Recent history shows that the political parties that have skillfully used big data and big data analysis as their campaigning tools have fared much better than parties that have depended on traditional forms of political campaigning and focus-group type consulting.

By using custom-created proprietary software and algorithms, we can create a template for the most effective messages that can be delivered to your block, city, county, district, state, region and country. Our software can pinpoint those issues that are on the minds and in the hearts of the voters in your area of concern or concentration.

Our system can be tweaked to reflect timely issues that can change exceedingly quickly. It can also easily be scaled to meet the needs of small local campaigns or large national ones. The insights we can glean never become stale or out of date because they are constantly being drawn from real-time data.

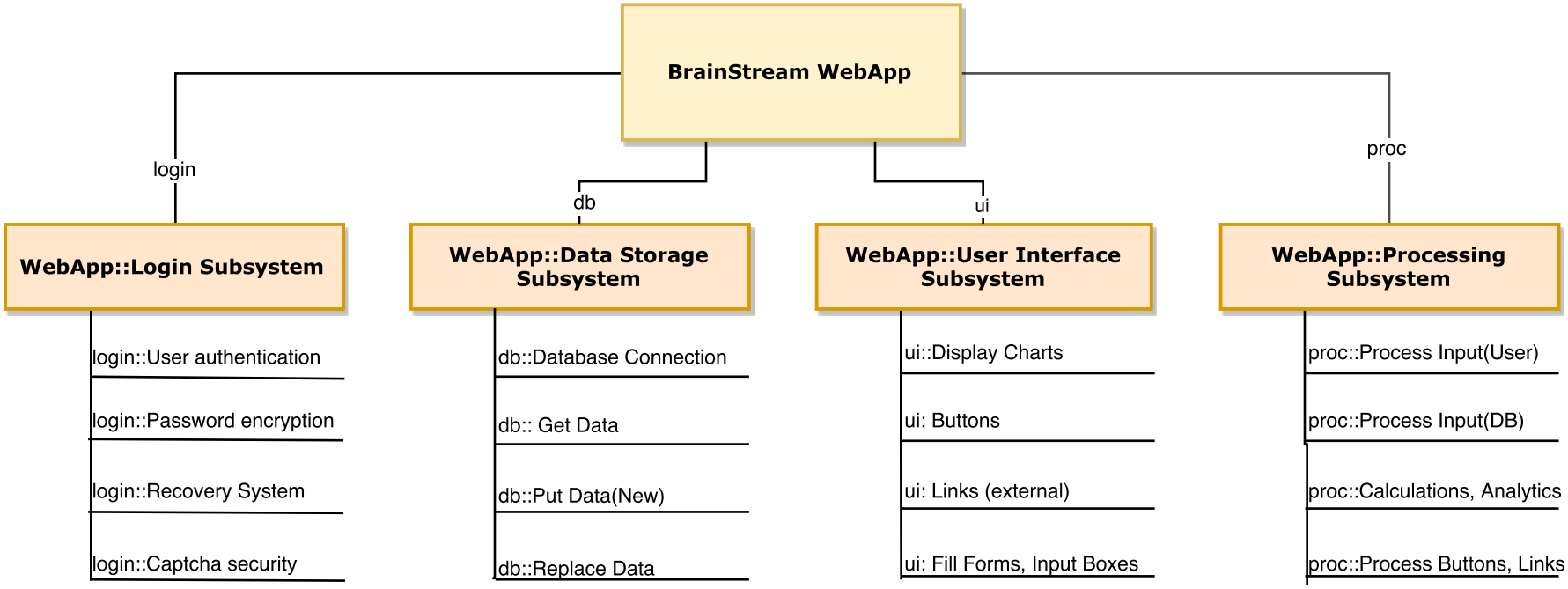
Our product can identify, find and drive your voters to action. Through the real-world knowledge our product provides we can help you create a winning campaign by knowing your electorate better and achieving greater influence while lowering overall costs.

All interpretive tools and documentation are included and further consulting services can be provided at additional fees.

Stake Holders

- **Candidates**
- **Political Party**
- **Campaign Staff**
- **Field Workers**
- **Electorate**

4. IDENTIFY SUBSYSTEMS



6 – Interfaces

User		
Field Name	**Data Type**	**Data Shape**
userID	Int	xxx
LastName	varchar	xxxxxx
FirstName	varchar	xxxxxx
Email	varchar	xxxx@xxx.xxx
typeOfUser	varchar	xxxx
relevantRegion	int	xxx
activeCampaign	int	xxx
Region		
regionID	int	xxx
regionName	varchar	xxxx xxxx
regionAddress	varchar	xxx xxx xxx
regionCity	varchar	xxxxx
regionPostCode	varchar	xxxxx
regionTopSearch	varchar	xxxxx
regionSecondSearch	varchar	xxxxx
regionThirdSeach	varchar	xxxxx
Campaign		
campaignID	int	xxx
campaignName	varchar	XXXX
campaignHead	varchar	xxxx
dateStart	varchar	xx-xx-xxxx
dateEnd	varchar	xx-xx-xxxx
campaignRegion	int	xxx
campaignBudget	Float	xxxx.xxxx
Speeches		
speechID	Int	xxx
speechVoice	audio	mp3
speechVideo	video	avi
speechUser	int	xxx

7 – Functional Requirements

- **The Database**: Store Data such as:
  - User information such as name, address, position
  - Search Histories based on demographic data.
  - Data about current and past campaigns
  - Logs
- **Front-End Framework:**
  - Display the webpages, buttons, forms, etc..
  - Display the Graphics and Reports it receives from back-end.
  - Create functionality for all the intractable UI.
- **Back-End Framework:**
  - Crawl through external webpages, polls, news articles, google searches, databases, etc. to pull out data relevant to the area searched.
  - Make judgements based on all the data received. Use mathematical algorithms to determine hot topics and flashes. Categorize data effectively.
  - Connect to the Database to store and collect data.
- **External Login System**:
  - Handle Username/Password data.
  - Authenticate data entered and show error messages
  - Authenticate Person Position, Authority.
  - Handle changes to members.

8 – Non-Functional Requirements

- **Performance:** Software is Web based. Response time on simple, not data-intensive pages predicted to take 0.5-1 second to load. Pages with lots of information predicted to take 2-5 seconds.
- **Reliability:** System may go down for 15 minutes every week during lowest traffic time every week. Its predicted that another 15 minutes may be spent per 3 months on un-expected failures and bugs. The recovery time is expected to be about an hour.
- **Capacity:** As the system is web-based, it has to be able to handle all incoming network traffic from any and all locations. However, U.S Bureau of Labor Statistics show that an average of 11 to 12 thousand government officials work during a presidential campaign, maxing out at most 20,000. Therefore the systems original size will be 20,000 and will expand to 100,000 with no major modifications.
- **Security:** Accounts will be password locked and passwords will be encrypted in the database. Extra measures of security will include security questions, lockouts from too many attempts, and password change recommendations.
- **Portability:** Webpage will be designed to run on all browsers and operating systems. It will also be able to run on portable devices.

9- Select a Process Model

Pick: Agile-EP

About Agile:

Agile is mostly described as an iterative waterfall model. This is because the software is delivered in iterations as opposed to delivering it as a whole. Software is developed in Sprints that last from 1 week to 4 weeks, and then presented and evaluated with the client before proceeding to the next sprint. This way, the changes or errors in development can be done much earlier, saving alot of money from post-delivery maintenance.

Why choose Agile for Project:

- Made mostly for small groups working in a tight environment on a single project.
- Extreme Programming allows building a software without knowing what the client wants.
- Less prone to errors and bugs after delivery important when time and effort after delivery is not possible.

10 – SAFETY CONCERNS

Safety Concern #1: Database Compromised

Implications:

- Hacker can access database from an external network
- Hacker can add and delete data from database including entire table
- Hacker can steal user sensitive information such as full name and address.

Defense:

- Symmetric Database Encryption
- Hashing sensitive data
- Transparent Data Encryption

Safety Concern #2: Denial-Of-Service Attacks

Implications:

- Hacker floods the network with requests
- Hacker crashes the server and website

Defense:

- Create proper Firewall protocols.
- Leave extra bandwidth to handle stress
- Monitor Inbound traffic for spikes.

Safety Concern #3: Cross-site scripting

Implications:

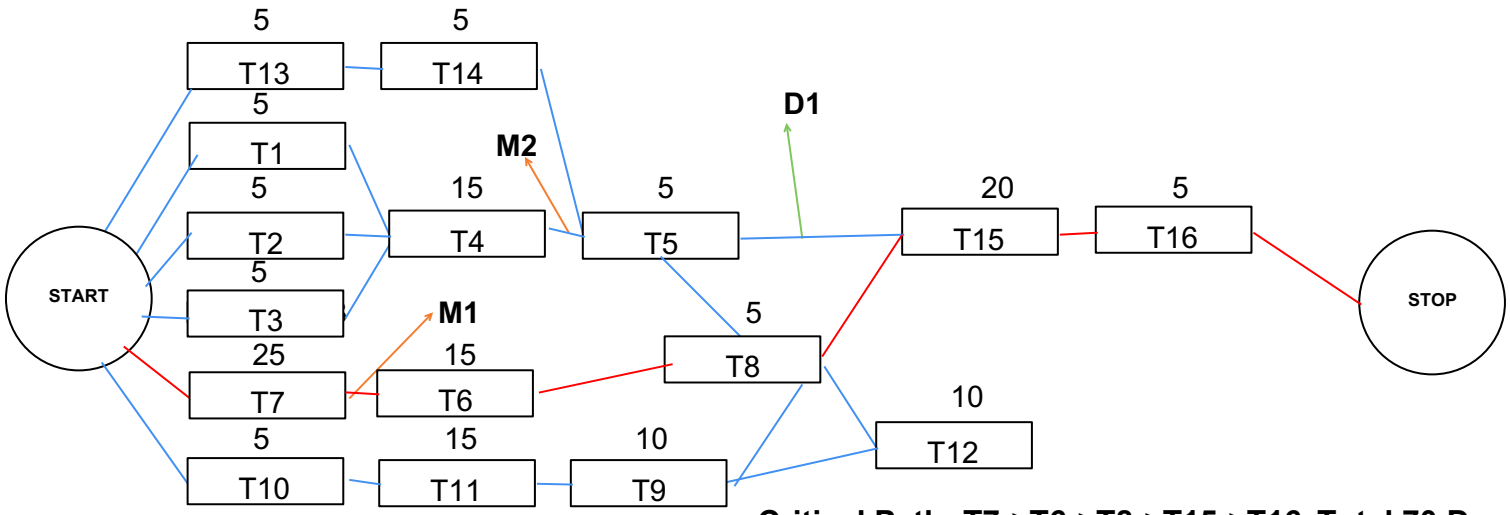
- Hacker can run scripts on the website
- Script can bypass safety measures
- Script can attack the database and back-end systems
- Script can disrupt or stop network traffic.

Defense:

- Contextual output encoding/escaping of string input
- Safely validating untrusted HTML input
- Flagging Suspicious Cookies

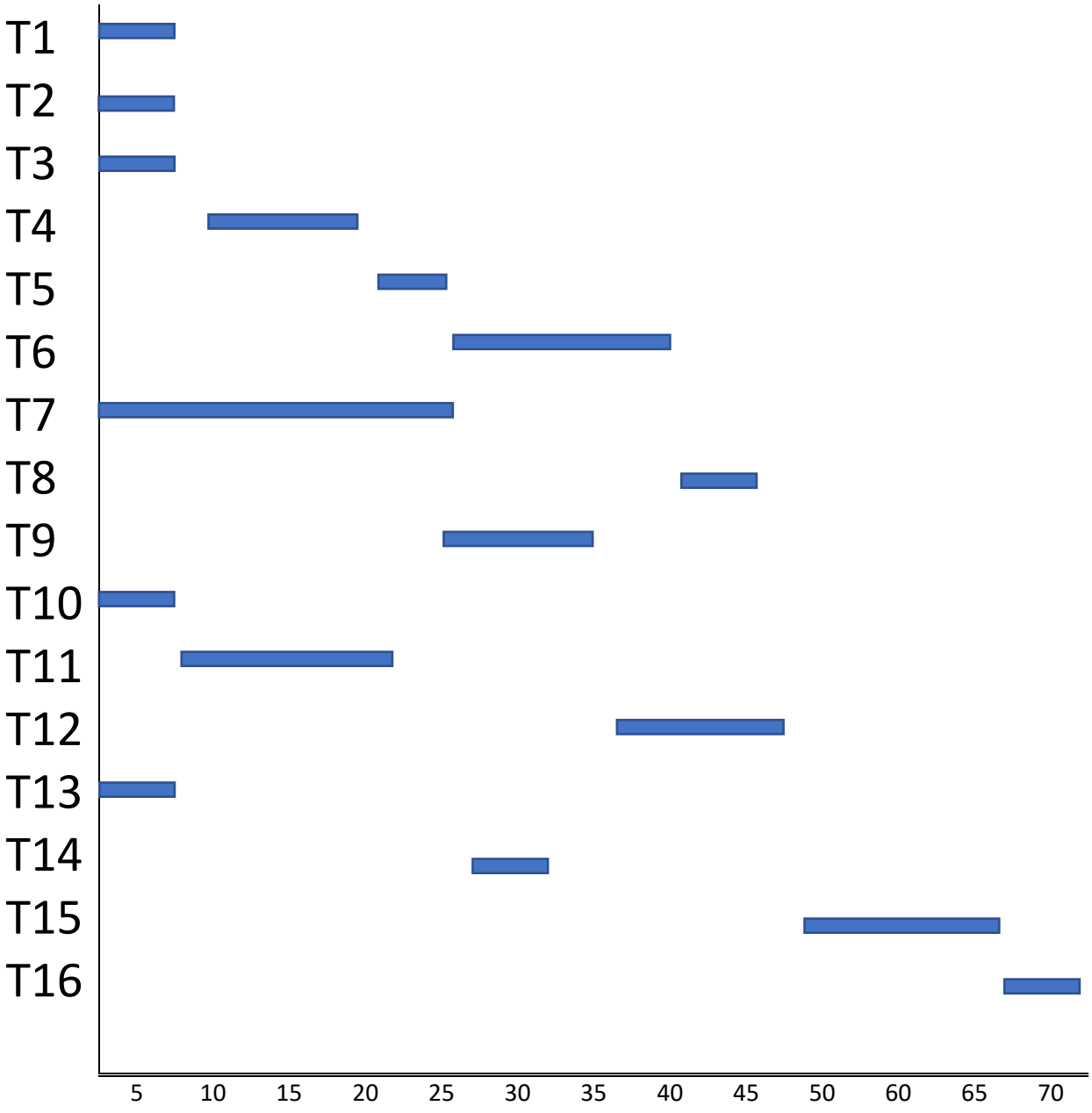
11 & 13 - ACTIVITY CHART, MILESTONES, DELIVERABLES, CRITICAL PATH

Task	Description	Duration (days)	Dependencies
T1	Create Tables for User Data	5	
T2	Create Tables for Campaign Data	5	
T3	Create Tables for search data	5	
T4	Setup Table Relationships	15	T1,T2,T3
T5	Setup Database Connection	5	T4
T6	Code Script to Crawl for Data	15	T7
T7	Create Algorithm to process crawled data	25	
T8	Process algorithm on data and setup organization of data in db	5	T5,T6,T7
T9	Code script to process user input on webpage	10	T11
T10	Design the webpage outlook	5	
T11	Code the pages according to design	15	T10
T12	Code out display of reports and charts and figures from retrieved data	10	T8,T9
T13	Setup connection with 3rd Parties	5	
T14	Code authentication of user using 3rd party softwares	5	T13, T5
T15	Create Settings Framework(Back-End)	20	T5,T8,T9,T14
T16	Code Encryption, Analytics, and maintenance code	5	T5,T9

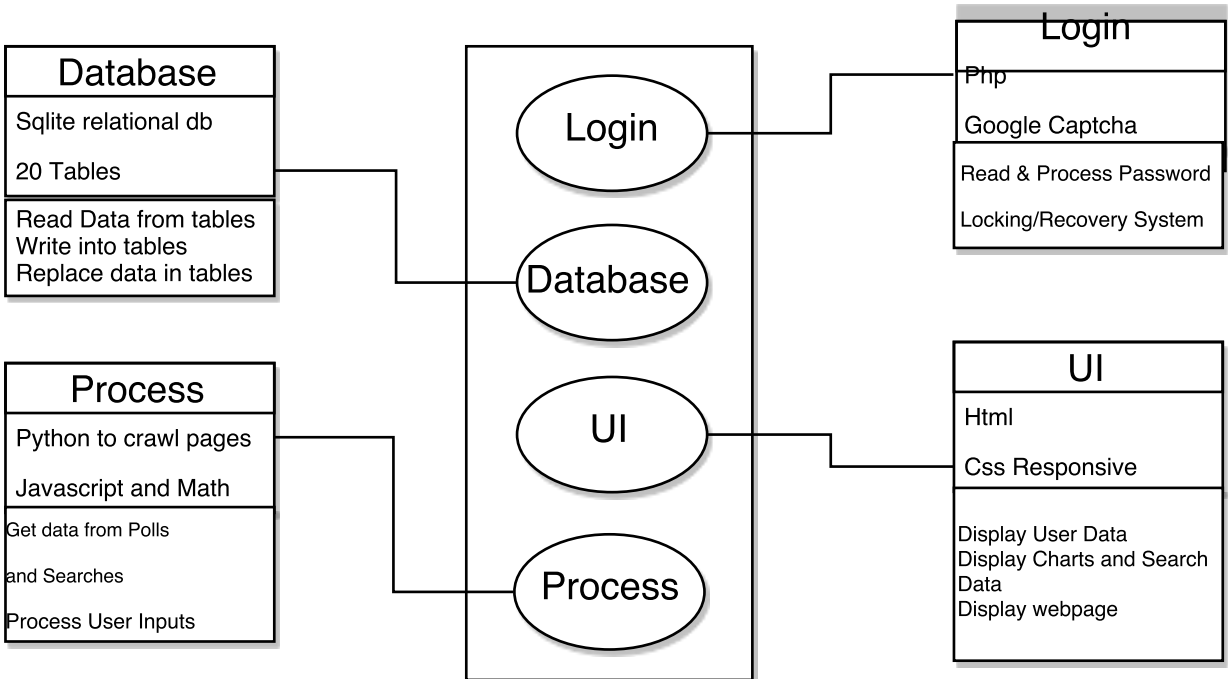


Critical Path: T7->T6->T8->T15->T16. Total 70 Days

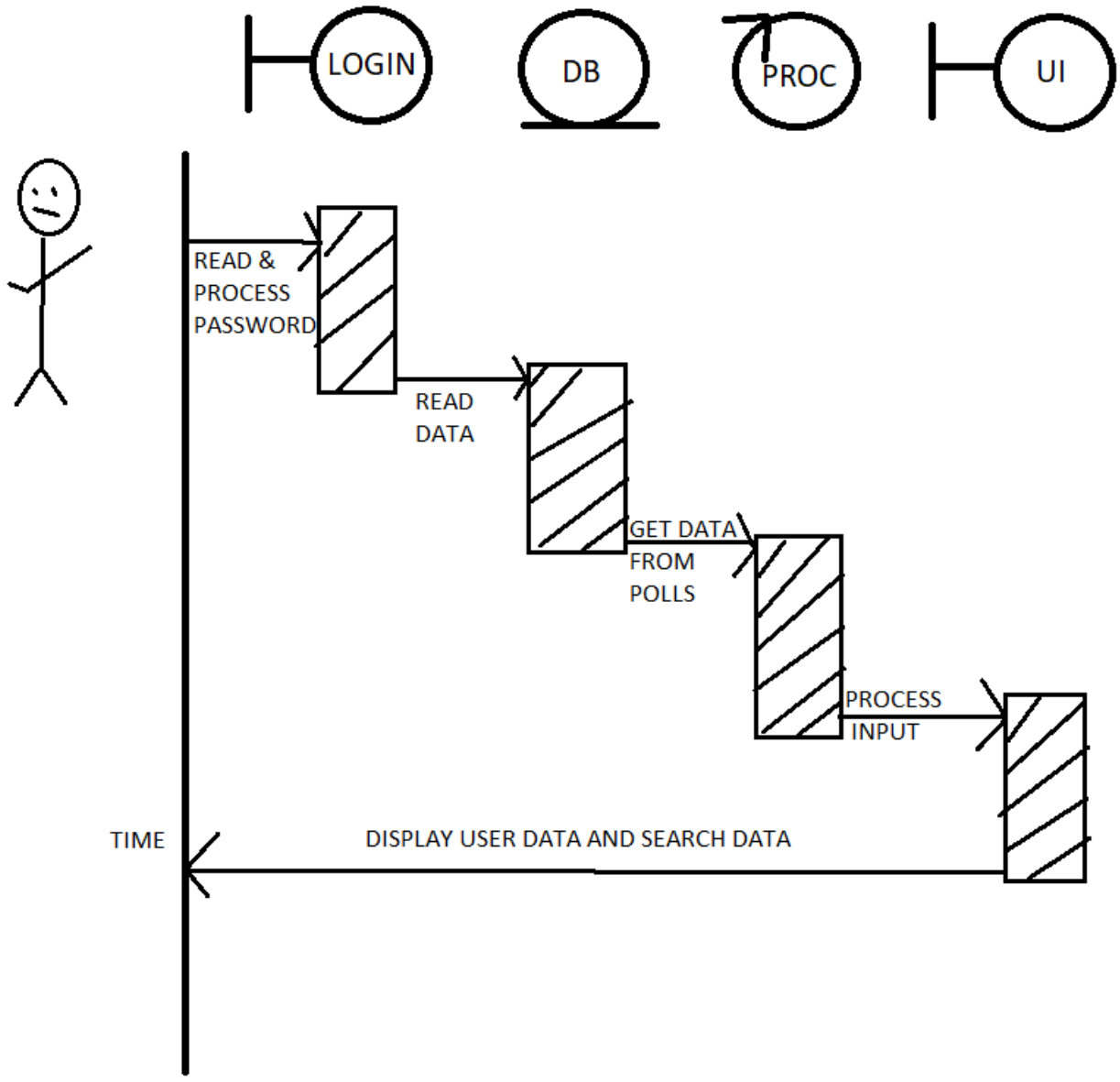
12 – BAR CHART


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
T13
T14
T15
T16
5
10
15
20
25
30
35
40
45
50
60
65
70

#14 - UML


Database
Sqlite relational db
20 Tables
Read Data from tables
Write into tables
Replace data in tables
Process
Python to crawl pages
Javascript and Math
Get data from Polls
and Searches
Process User Inputs
Login
Database
UI
Process
Login
Php
Google Captcha
Read & Process Password
Locking/Recovery System
UI
Html
Css Responsive
Display User Data
Display Charts and Search Data
Display webpage

15 - SEQUENCE DIAGRAM



17 – TEST PLAN

- **System:**
  - **Test Login System**
  - **Test Database Storage system**
  - **Test User Interface Systems**
  - **Test Algorithm and Processing/Calculations.**

- **Subsystem:**
  - **Login:**
    - **Test User Authentication**
    - **Test password encryption**
    - **Test password recovery system**
    - **Test security and authentication systems**
  - **Data Storage:**
    - **Test Proper DB Connection**
    - **Test Data Retrieval**
    - **Test Data replacement**
    - **Test Insert of new data and relationships**
  - **User Interface:**
    - **Test Charts and reports and figures**
    - **Test all buttons**
    - **Test all hyperlinks**
    - **Test forms and all text inputs**
  - **Processing:**
    - **Test proper calculations**
    - **Test for proper calculation input**
    - **Test Functions on buttons and links**
    - **Test Algorithms used for calculation**

- **Unit (Login System):**
  - **U: Enter "John Smith"**
  - **S: Error "john smith" is not recognized**
  - **U: Enter "JohnSmith"**
  - **S: Ok. Enter Password**
  - **U: "*****"**
  - **S: Error "Must be longer than 8 digits"**
  - **U: "*********"**
  - **S: OK.**

18 – MAINTENANCE PLAN

Daily Maintenance

- Perform Backup(Automatic) of database and crucial sub-systems
- Update Plugins(Automatic)
- Check for spikes in Network
- Review any issues sent from Analytics, Security, or Email Servers

Monthly Maintenance

- Check if all pages load properly.
- Check for any broken links
- Check for 404 Errors.
- Check for any type of Console errors.

Monthly Maintenance

- Check website loading speed.
- Review security scans and resolve any issues
- Review website statistics and evaluate conversion rates.
- Review local visibility and search engine optimization.

Quarterly Maintenance

- Review website – what could be improved?
- Check for Outdated 3rd party libraries or software.
- Review and tweak meta title and meta description tags
- Test and tweak website to improve conversion rates.
- Test Website for browser compatibility.
- Check the uptime logs

Yearly Maintenance

- Update the copyright date in footer
- Review each page of the site for content accuracy
- Renew SSL certification if needed.
- Check backup health by restoring the most recent backup to a separate web server
- Validate your site. (CSS/HTML, Accessibility, Mobile-friendly)
- Review your website strategy to align with your business goals.

16 – WEB SITE CODE

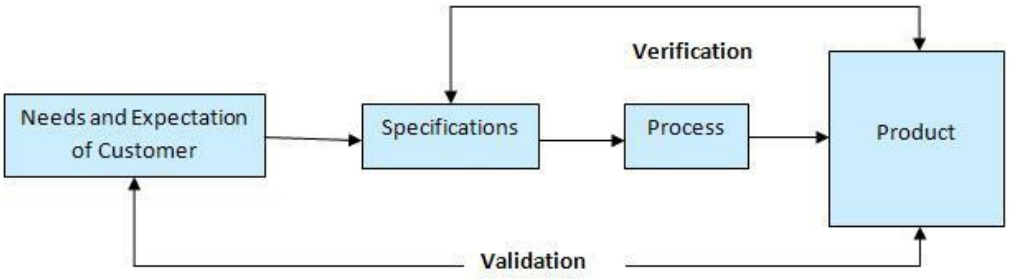
```
<html lang="en-US"><head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="chrome=1">

<title>CS350PROJEKT</title>
<meta property="og:title" content="CS350">
<meta property="og:locale" content="en_US">
<meta name="description" content="CS 350 PROJECT FALL 2017">
<meta property="og:description" content="CS 350 PROJECT FALL 2017">
<meta property="og:url" content="https://alikalkandelen.github.io/CS350/">
<link href="https://fonts.googleapis.com/css?family=Arvo:400,700,400italic" rel="stylesheet" type="text/css">
<link rel="stylesheet" href="https://pages-themes.github.io/dinky/assets/css/style.css?v=46349262f43f70d5dc78cf83775850b0466aa573" >
<script src="https://pages-themes.github.io/dinky/assets/js/scale.fix.js"></script>
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
<!--[if lt IE 9]>
<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<style>
header{
padding: 34px 80px 22px 50px;
}
</style>
</head>
<body>

<div class="wrapper">
<header>
<h1 class="header">BRAINSTREAM</h1>
<p class="header">CS350 PROJECT FALL 2017</p>
<ul>
<li class="download"><a class="buttons" href="combinepdf.pdf">Download PDF</a></li>
<li class="download"><a class="buttons" href="projectfiles.zip">Download ZIP</a></li>
```

```
</ul>
<p>Ali Kalkandelen <br> Samuel Owusu-Biney <br> Stephen Kilnisan</p>
</header>
<div id="results" class="hidden"></div>
<section id="example1">
HELLO WORLD
</section>
<script src="pdfobject.min.js"></script>
<script>
var options = {
pdfOpenParams: {
pagemode: "thumbs",
navpanes: 1,
toolbar: 1,
statusbar: 0,
view: "FitV"
}
};
var myPDF = PDFObject.embed("project.pdf", "#example1", options);
var el = document.querySelector("#results");
el.setAttribute("class", (myPDF) ? "success" : "fail");
el.innerHTML = (myPDF) ? "" : "The PDF embed Didn't Work. Please try another browser or deleting your cache and trying again.";
</script>
</div>
<!--[if !IE]><script>fixScale(document);</script><![endif]-->
</body>
</html>
```

20 - VERIFICATION AND VALIDATION



	Validation	Verification
Questions:	• Does the final product meet the business needs of the customer? • Does the product meet all the testing requirements?	• Does the product according to the specifications? • Does the implementation meet the design? • Does the product follow the proper cycle?
To do:	• Execute the software • Test executed software (alpha,beta,FAT) • Validate product with business	• Review specifications and implementation, making sure nothing is missed. • Review the Requirements and match it with the product

21 - COCOMO

Nominal Effort: $3.2 \times (KDSI)^{1.05}$

Estimated DSI = 4000

$3.2 \times (4)^{1.05} = 13.72$

(1.15)(1.08)(.85)(1)(1)(1)(.87)(1.19)(.91)(.86)(1.1)(.95)(.82)(.91)(1.1)

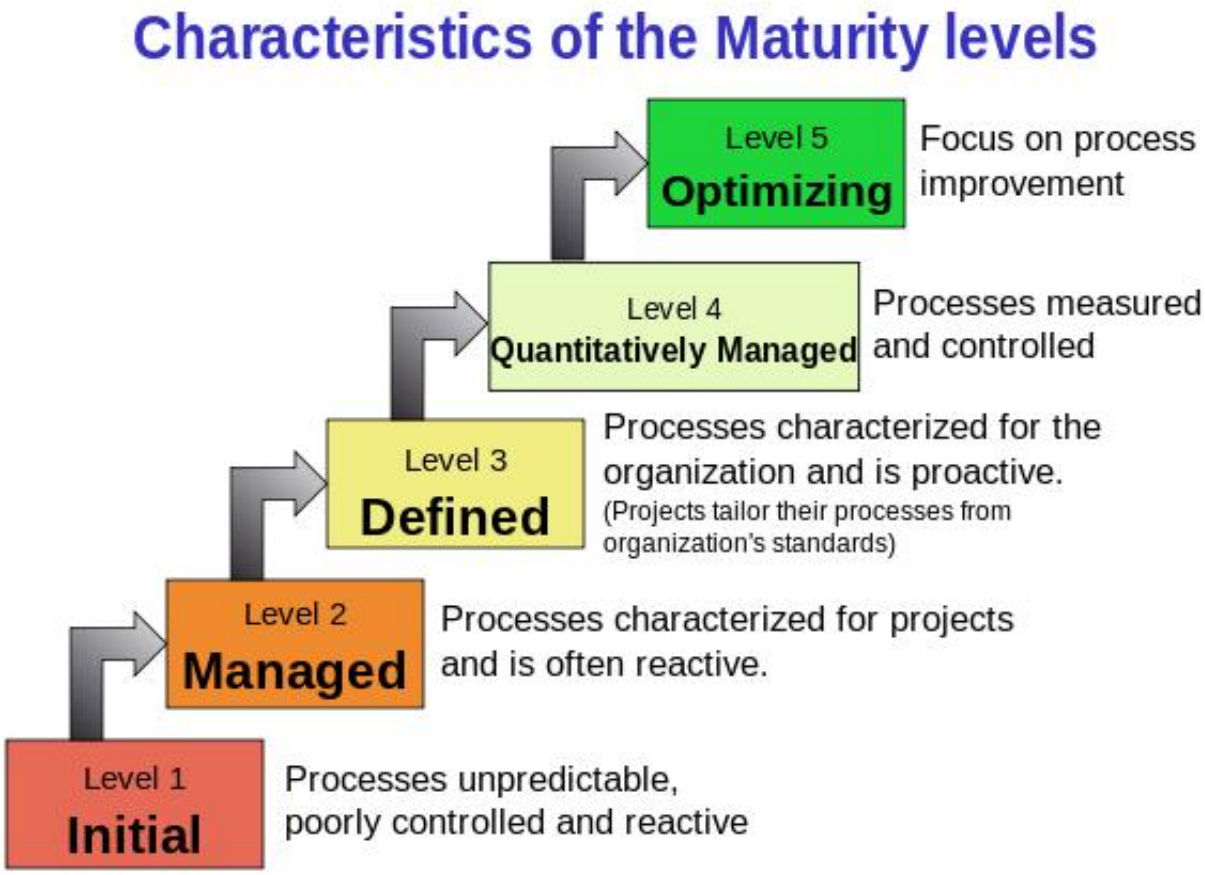
= .74

13.72 x .74 ~= 10 Person-Month

22 - Decommisioning Plan

- Determine Retirement Strategy
    - Is the program in full retirement or migrating to a newer system?
    - Will the retired program be backed up or purged forever?
    - How many people are involved in the current system?
    - Size of Database? Size of Code? Size of Traffic?
- Database:
    - Database Refactoring to newer software, or complete, safe deletion if full retirement.
    - Backups are deleted or migrated depending on the decision
- Documentation:
    - Docs are updated to include decommissioning portions and plans
- Users:
    - Users are informed of migration or removal.
    - Users are directed to a new system in a seamless manner
    - Users receive proper training into understanding new system
- Implementation:
    - Remove all code from server.
    - Delete proper files and folder structures.
    - Backup to another PC if needed
- Business:
    - Decommission costs determined.
    - Staff is properly dismissed or repositioned
    - Business approves plans and budget

23 - CAPABILITY MATURITY MODEL



CMM LEVEL PICKED: **LEVEL 2 - MANAGED**

Reasons:

- The characterization of projects are well defined
- Processes for the project are clear and explanatory.
- Software is from a brand new group, so no organizational standards are set.
- Processes are defined but not well controlled.
- Processes have no definition for improvement or proactive for the future.

COST PER PERSON : $20,000 Per Month Per Person.

Based on 10 person month * 20K = **$200,000 For Project.**

24 - DATED LOG

Person Name	Person Position	Description	Date
Alan Smith	Operations Manager	Set-Up Orgnization structure and personnel	09/01/2017
John Adams	Database Analyst	Made priliminary database structures	09/20/2017
John Doe	Programmer	Tech stack setup and analysis	09/22/2017
Jane Doe	Senior Analyst	Major change in life cycle method	10/10/2017
Sam Silverstein	Programmer	Changed tech stack to accomodate project	10/20/2017
Tom Turner	Project Manager	Added developers to project to speed up prod	10/31/2017
Adam almosa	Assistant VIP	provided additional funding to proj	11/01/2017
Laurie Comes	Secretary	Made updates to office space	11/02/2017