Alia ElKattan
Heorhii Skovorodnikov

Online Publication Website

Deliverable #2

Software Requirement Specifications

Description:

A group of students on campus are working on an online publication called Multiplicity, where seven students and alumni are writing pieces about their experiences of how growing up with the internet has impacted different aspects of their lives, including identity, politics, relationships, and communication. They are looking for this article series to be hosted on an independently-run website that reflects the uniqueness of the work and is visually strong and interactive, and so did not want to depend on usual Medium blogs or website builders. As such, we agreed with them to build their website as part of our software engineering project, as we thought the experience of working with a “client” would help us put a lot of the lessons we learn in class to practice.

Gathering requirements

Since there is actually a group outside of us who we are building this website for, we had to be in constant collaboration with the team regarding what they want their websites to look like. As part of the process, we had conversations with them on what the articles were about, to understand the general purpose and feel of the project, then spoke with the project manager and designer on what specific functionalities they would like to include. This included several conversations and a back and forth in terms of presenting to them examples of websites and seeing how what they want is similar or different. In addition, both of us spent time looking at other websites and gathering a list of the requirements they implemented, and contrasted it to the information we have from our conversations with our own “clients” to come up with our list.

Functional requirements

User Interface Requirements

- New articles must be able to be inputted through an easy-to-use template for writers to access (no dealing with code on their side)
- There needs to be a navigation bar to take the user to different pages
- Writers can embed their own html/css code
- Ability to sort/filter through the articles

Alia ElKattan
Heorhii Skovorodnikov

Design Requirements

- Articles should be able to accommodate text, images, videos, or animation
- Website must be fully mobile-friendly and responsive

Business-related requirements

- User to be able to share to social media platforms
- User can subscribe to the mailing list through the front page
- Site admin must be able to track site analytics and viewership

Non-functional requirements

- User should be able to intuitively understand how to navigate the website and access articles
- Mobile-friendly version of the website should look mobile-native and adjust content presentation to fit a smaller screen
- Website layout should encourage readers to check out more articles and spend more time on the website
- Website should be interactive and engage the users beyond simply consuming/reading content

Alia ElKattan
Heorhii Skovorodnikov

Feasibility Study

Technical Feasibility:

1) Users' and analysts' familiarity with the business area: Users are writers that had experience writing for the online magazines/publications. The team's members are well familiar with web development technologies required to produce the product, templates that can provide design ideas as well as potential additional technologies will be learned in the process of development.

2) Familiarity with technology:
   The technical tools we are going to use:
   - Web dev frameworks like React and Vue
   - Web dev languages like HTML, CSS, Javascript
   - Database management languages SQL
   - Runtime environment Node.js
   - Database management MongoDB
   - Website hosting on Github hosting
   - Website and web app templating with Bootstrap

Team is mostly familiar with web dev tools except database management and SQL as well as web dev frameworks like React and Vue.

3) Project Size:
   It is about 2 persons for 3 months.

4) Conclusion:
   Potential risks include unexpected difficulty in learning the technologies required and change of technology tools required if the current tech stack will not integrate well.
   The risk in this stage is low due to the team's decent familiarity with the technology and the business area.

Alia ElKattan
Heorhii Skovorodnikov

Economic Feasibility:

As the publication will be initially designed to be non profit and the team volunteers their time and effort for the development most of the values currently will be 0, potential income will be donations and current expenses are hosting and domain name.

Costs	**Period 1**	**Period 2**	**Period 3**	**Total**
Salaries	0	0	0	0
H/W & S/W	0	0	0	0
Training	0	0	0	0
Support & maintenance	80	0	0	80
Total Costs	80	0	0	80
Benefits	0	0	300	300
Total benefits	0	0	300	300
NCF	(80)	0	300	520
CNCF	(80)	(80)	220	60

Values in ones of AED:
NCF: Net Cash Flow
CNCF: Cumulative Net Cash Flow
One period corresponds to one month
H/w and S/w correspond to Hardware and Software respectively

- ROI

  (300-80)/80 = 2.75%

- BEP

  (300-20)/300 = 0.27 = 27%

Conclusion: ROI is good for non-profit and BEP is good as well, risk is low.

Alia ElKattan
Heorhii Skovorodnikov

Project Estimation/Sizing

Functionality	Input	Output	Queries	File	Program Interface
Search / Navigation	1	1	1	6	1
Publishing	1	1	1	1	1
Sharing	1	1	0	0	1
Analytics	1	6	1	1	1
Code Embedding	1	1	2	2	1
Mailing List	1	1	0	0	1
Article Sorting	1	1	1	0	1
Database Fetching	1	1	1	1	1
Login	1	2	3	1	1
Database Upload	1	0	1	1	1

Alia ElKattan
Heorhii Skovorodnikov

	Complexity				
Description	Total #	Low	Medium	High	Total
Inputs	**10**	**4*3**	**4*7**	**2*6**	**52**
Outputs	**15**	**4*4**	**6*7**	**5*6**	**88**
Queries	**11**	**8*3**	**3*7**	**0*15**	**45**
Files	**19**	**14*5**	**5*10**	**0*15**	**120**
Program Interface	**10**	**6*5**	**4*7**	**0*14**	**58**
Total Unadjusted Function Points (TUFP)					**366**

Total processing complexity (PC):

Tasks	Complexity (0-3
Data communication	3
Team cohesion	1
System responsiveness	3
Design implementation	2
Total processing complexity (TPC) =	9

Adjusted processing complexity (APC):
APC = 0.65 + (0.01 * 9) = 0.74

Total adjusted function points (TAFP):
TAFP = 366 * 0.74 = 270.84

Alia ElKattan
Heorhii Skovorodnikov

Converting Function Points to Lines of Code (LOC):

Language/Tool	Number of Lines of Code/FP
HTML/CSS	100
JavaScript (front-end)	200
JS/node (back-end)	50

- 29% will be in HTML/CSS development
- 57% will be JavaScript programming
- 14% will be building the back-end infrastructure

Number of lines of code (LOC) = TAFP * # of (LOC/FP) * %

HTML/CSS = 270.84 * (100) * (29/100) = 7,854
JavaScript = 270.84 * (200) * (57/100) = 30,875
Back-end = 270.84 * (50) * (14/100) = 1,895

Estimating the effort:

Effort = 2.4*1895/1000 = 4.6 person month

Estimating the schedule time:

Time= 2.5*(4.6)^0.38 = 4.47 months

Estimating the number of persons:

Average of # of persons = effort/time = 4.6/4.47 = 1.03 persons

Since it would take 1 person estimated around 5 months to complete the project, this estimate means that it would take us, as 2 people, the required 2.5 months to complete it.

Alia ElKattan
Heorhii Skovorodnikov

Project Development Work Plan (Gantt Chart)

Task Name	Week 1: March 20	Week 2: April 5	Week 3: April 12	Week 4: April 19	Week 5: April 26	Week 6: May 3	Week 7: May 10
Setup GitHub repo and find template							
Identify system requirements							
Build website skeleton/directory							
Complete website navigation							
Develop a template for inputting articles							
Add capability for sharing to social media							
Add ability to track website analytics							
Complete website front page							
Complete website article pages							
Add feature to sort through articles							
Conduct continuous cycles of user testing							

The Software Process Model used will be an Incremental Model (specifically a modified agile model), as it will allow us to frequently share status updates with our stakeholders, receive feedback, and make sure we are building the website that suits all the right needs. We will be working on shortened 1-week sprints (instead of the usual 2-4 in agile) to suit the short nature of this project, and will check in with each other as we have been doing on progress and plants for the coming sprint. The outline for the tasks to be completed during the sprints can be seen in the Gantt chart below.