Learning Report – Emotion Detection and Banking System

GLOBAL ENGINEERING ACADEMY

GENSIS

L&T Technology Services

Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	Approved By	Remarks/Revision Details
1	17/09/2020	Shabana R P			
2	18/09/2020	Shabana R P			
3	19/09/2020	Shabana R P			

Contents

Table 1

ACTIVITY 1 System/Software Development

1.Introduction:

This project proposes a new algorithm for emotion classification using NLP, that requires fewer data for training. Instead of using words and word relation i.e. association rules from these words are used to derive feature set from classified text documents.

2. Product Definition:

An emotion is a feeling such as happiness, love, fear, anger, or hatred, which can be caused by the situation that you are in or the people you are with.Emotion can be expressed in many ways that can be seen such as facial expression and gestures, speech and by written text.

3. SWOT Analysis:

Strength:

Efficient

Accuracy

Easy to distinguish the documents based on categories

Weakness:

Time consuming

Syntactically similar works are also emotionally similar

Opportunity:

Extending dataset Using word2vec tool Adding POS Taggers to Emotion words

Threats:

Extending dataset should be done carefully

4. Requirement Gathering

Research: aging and costing in terms of performance and accuracy

Years	Performance	Cost value(accuracy)
2002	2.5 million emotions tweets covering 7 emotion categories for automatic emotion detection	60.4%
2012	Unigram method, Lemmatized unigram, Naïve Bayes lexical model	65.57%
2014	LDA and SVM	70%

Table 1:Requirement Gathering 1

Past:

Keyword-based:

In Existing solution method detection of emotions is limited to some short documents. They detect emotions by classifying it into positive , negative and neutral.

Limitations:

Ambiguity in Keyword Definitions
Incapability of Recognizing Sentences without Keywords Lack
of Linguistic Information

Present:

Sentiment Analysis (SA) or Opinion Mining (OM) is the computational study of people's opinions, attitudes and emotions toward an entity. The entity can represent individuals, events or topics. These topics are most likely to be covered by reviews. The two expressions SA or OM are interchangeable.

In [1], the authors explore the field of sentiment analysis. According to them domain-specific corpus gives better results than working on the domain independent corpus. There is still lack of research in the field of domain-specific SA which is sometimes called context-based SA. This is because building the domain-specific corpus is more complicated than using the domain-independent one

Future:

Above project can be fine-tuned by implementing it with POS tagger and word2vec tool by making it domain independent

Added features:

POS Tagger
Word2vec tool
Comparing accuracy of keyword based and POS Tagging based methods

4 . High level and low level Requirement(traditional):

4.1 High level requirement

1) Data Pre processing
2) Identify basic emotions
3) Extend dataset

4.2 Low Requirement:

1. Data Preprocessing
   - ❖ Tokenization
   - ❖ Lemmatization
   - ❖ Stemming
   - ❖ Stopword removal

2. Identify basic emotions
    - Classify emotions by implementing algorithms
3. Pos tagging
4. Classification
5. Extend Dataset
    - Implement Word2vec tool

4 . High level and low level Requirement(agile): High Level Requirement:

ID	Description
1	4) Data Pre-processing
2	5) Identify basic emotions
3	6) Extend dataset

Table 2

Low Requirement:

ID	Description
1	Tokenization: Tokenization is a key (and mandatory) aspect of working with text data
2	Lemmatization: takes into consideration the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma.
3	Stemming: algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word.
4	Stop word removal: A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore
5	Classify emotions by implementing algorithms :Classify them into basic emotion classes
6	Pos tagging: is the process of assigning a **part-of-speech** like noun, verb, pronoun, preposition, adverb, adjective or other lexical class marker to each word in a sentence. The **POS tagger** assigns to each token in the input one of **POS tags**.
7	Implement Word2vec tool: To extend database implement word2vec

Table 3:Low level Requirement

Design Models:

a. High-level design

Structural Diagrams:

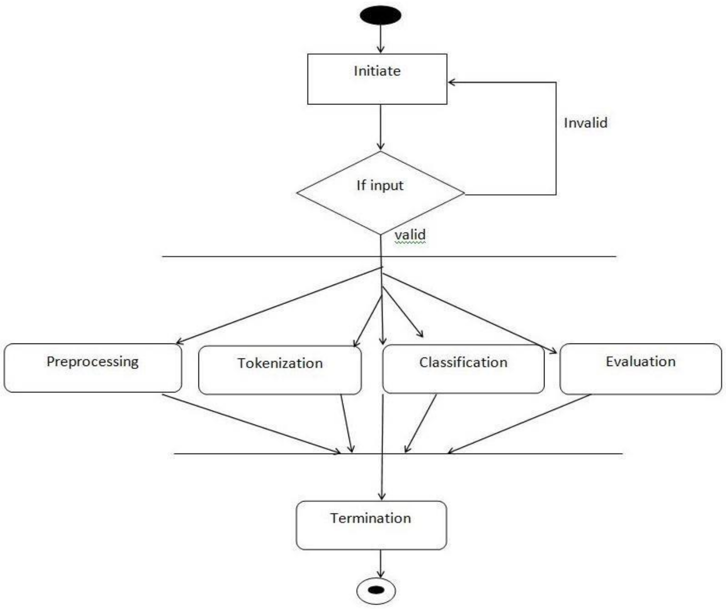


Figure 1:Emotion classifier

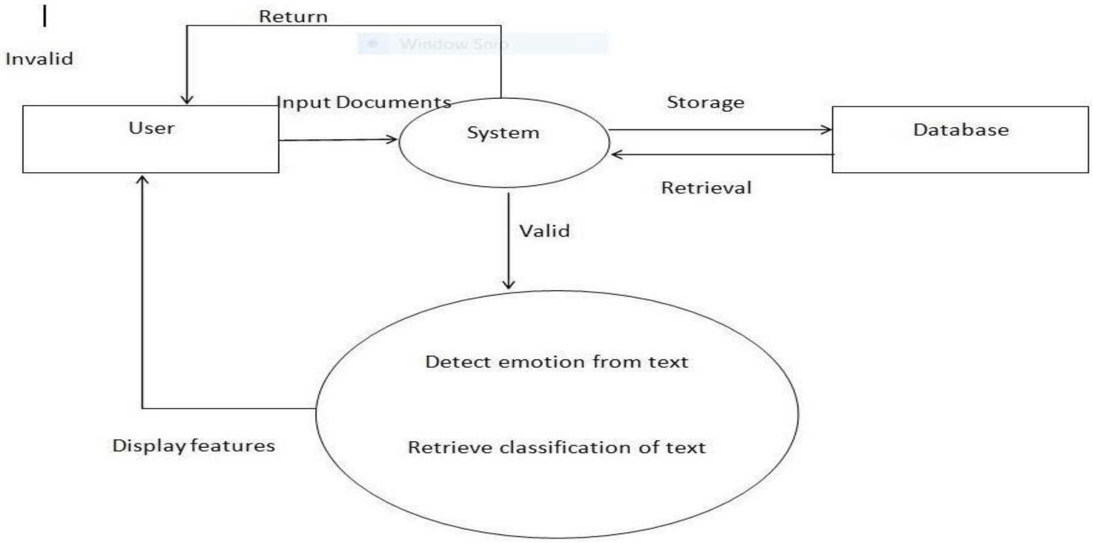


Figure 2:information retrival

Behavioral diagrams

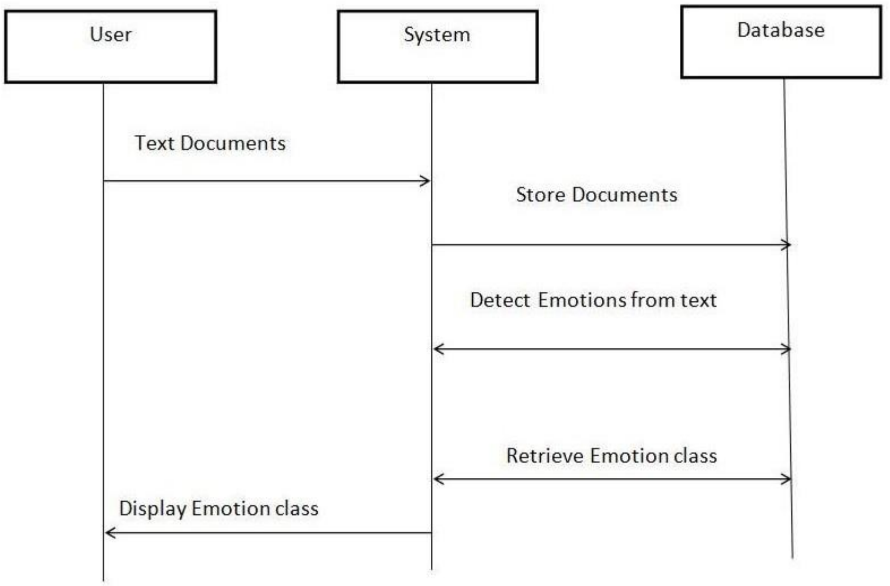


Figure 3Sequence diagram for classifying emotions

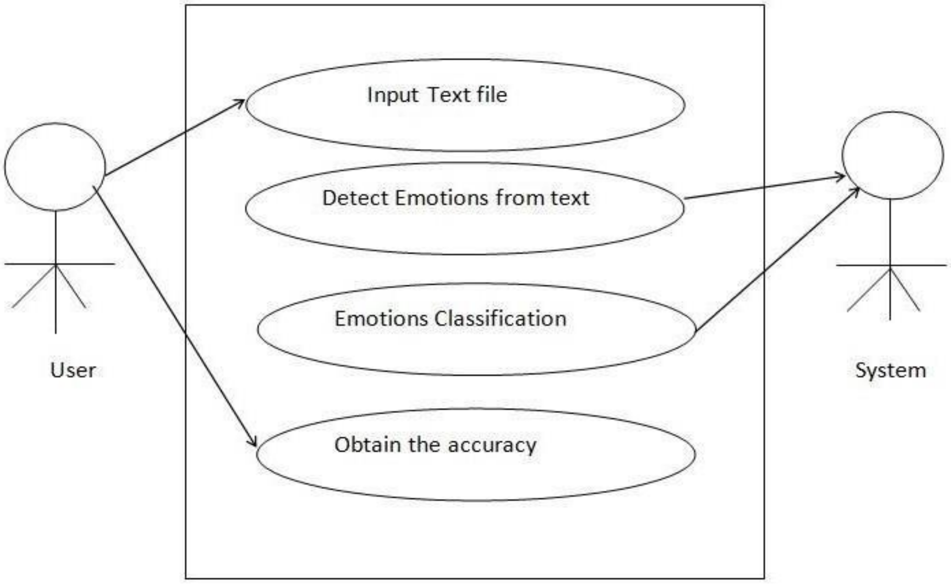


Figure 4:Use Case Diagram for obtaining accuracy

b. Low-level design

(1) Structural Diagrams:

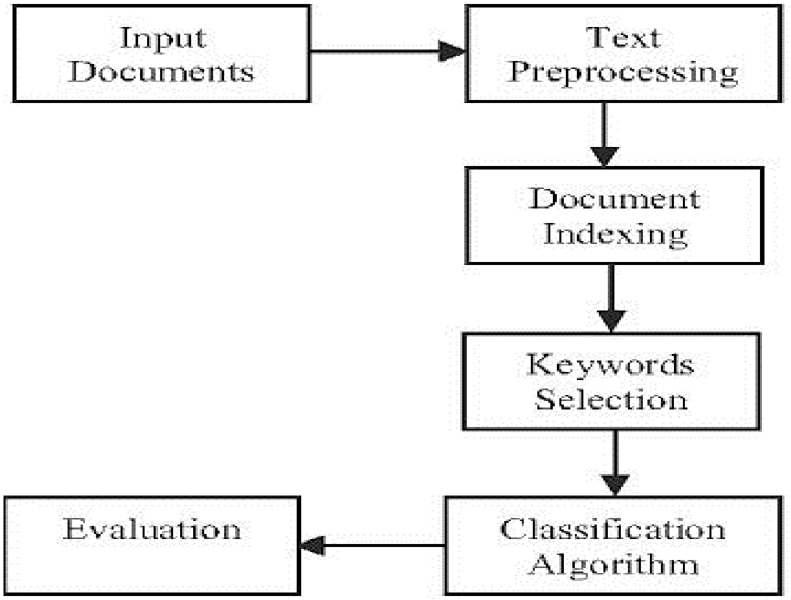


Figure 5:Feature selection

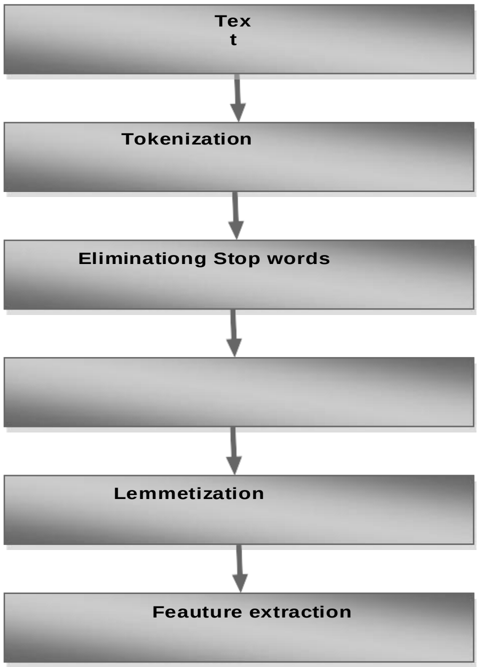


Figure 6:Process

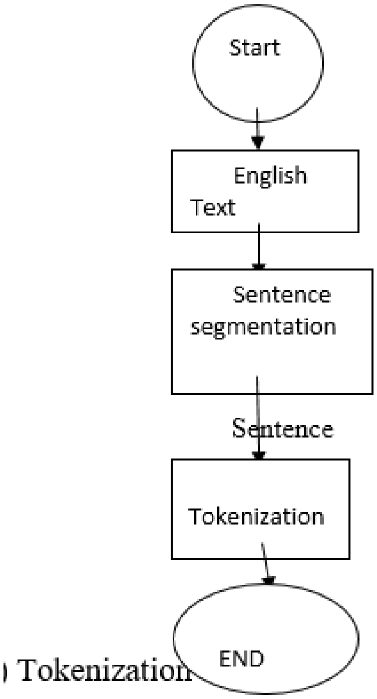


Figure 7 Tokenization

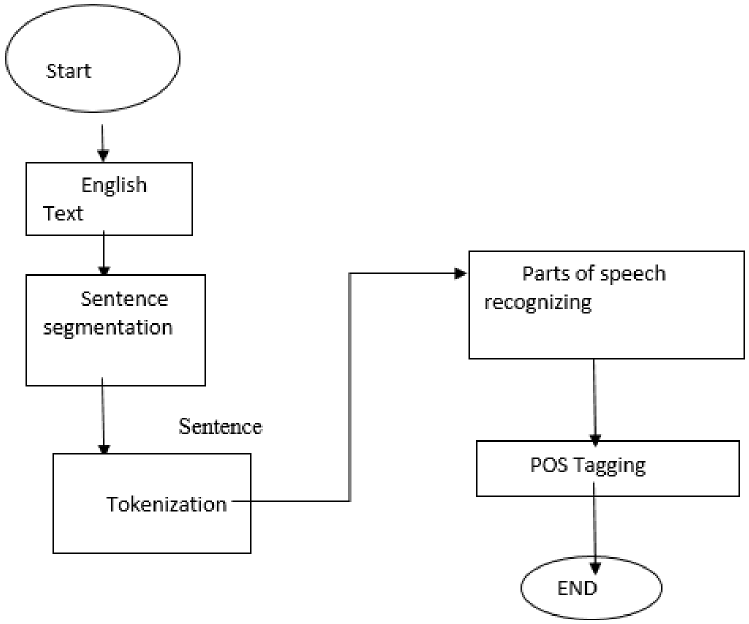


Figure 8:Pre processing

Test Plans (agile method):

Requirement Based Test cases: To classify sentences into basic emotions

ID	Description	Pre condition	Expected input	Expected output	Actual output
1	Classify the sentence into fear Class	It should identify fear keyword with word much	I am having so much fear about my life	Emotion Detected: Fear	
2	Classify the sentence into joy class	It should identify happy, good keyword with word very	I am very happy and good	Emotion Detected: Joy	
3	Classify the sentence into sad class	It should take intensity of sadness	I was sad by hearing that news	Emotion Detected: Sad	
4	Classify the sentence into Anger class	It should take intensity of sadness	The sport riot caused too much anger in players and fans	Emotion Detected: Anger	
5	Classify the sentence into disgust class	It should take intensity of sadness	It was a disgusting move by the government	Emotion detected: disgust	

Table 4 Requirement Based Test cases

2. Scenario Based Test cases:

ID	Description	Pre-condition	Expected input	Expected output	Actual output
1	Keyword based approach:	If not is used before the emotion word or within some specified window it should be considered and sentence should be classified according to their Classes	I am not happy	Sad	
2	POS Tagging method:	If there are more tags with more emotional words in the sentence then adjective and adverb tagged words should be prioritized	I am not angry	Neutral	

Table 5:Scenario based

3. Boundary Based Test cases:

ID	Description	Pre-condition	Expected input	Expected output	Actual output
1	Positive or above neutral	: If different adjectives or adverbs are given before the emotional word it should detect the degree of intensity	I am very happy today	Happy It should consider adverb (very in this case) along with emotional word ie happy to depict the emotion and consider its intensity	

2	Neutral (nullifying)	If different adjectives or adverbs are given before the emotional word it should detect the	I am neither happy nor sad	Neutral It should nullify this statement to neutral as it contains Both positive and negative	
		degree of intensity		emotion	
3	Neutral	It should consider this sentence neutral as there is no emotional words in it.	Delhi is the Capital of India	Neutral	
4	Negative or below Neutral	If different adjectives or adverbs are given before the emotional word it should detect the degree of intensity	I found it very disgusting and I am angry	It should consider adverb (very in this case) along with negative emotional words i.e. disgusting and angry to depict the emotion and consider its degree of intensity. if we assign weights, we will have more weightage to disgusting than angry	

Table 6:Boundary Based Test cases

Activity 2-AGILE CONCEPTS

Theme:

Emotion detection from Text

- This project is aimed to design a prototype which classifies 5 different levels of emotions from smaller to larger text documents using NLP.
- To classify text as emotional or non-emotional text.
- To compare Natural Language processing.
- To develop a prototype to provide overall accuracy for 5 different emotions.
- To find out Parts of speech for tokenized word.
- To compare Accuracy of both POS and Keyword based Emotion detection methods.

Epics:

1. Data Pre processing
    - Tokenization
    - Stop word removal
    - Case folding
    - Stemming
    - Classifying Emotions
    - Detect and Classify Emotions into 5 different Categories.
    - Accuracy
2. Identify basic emotions
    - Classify emotional and non-emotional keywords and then apply algorithms on emotional text to detect correct emotions
3. Extend dataset
    - To include new keywords data set should be extended using word2vec-tool

User stories**:**

Pre-Processing (epic 1)
Tokenization In this process, divide the user query input into small tokens. that is divide the text or sentence into words.
EX. I am not happy today, I am feeling very sad
Tokens: [I] [am] [not] [happy] [today] Next is **Stop word removal**, In this process remove the stop word from the text or sentence, like a, an, the, after, before etc. EX: I am not happy today, I am feeling very sad In the above example after removing stop words the text should be like ANS: happy today feeling very sad, after removing stop words next step is **Case folding** In this process, convert the all words into lowercase for easy comparison. EX: happy today feeling very sad Last step is **Stemming**, In this process, convert the all words into root words. EX: feeling ANS: feel

Effort: 8 hours

Identifying Emotions(epic 2)
The system should be able to detect emotions from different sized documents. Emotions are divided into 5 different types based on the Paul Ekman Theory. Happy, Sad, Angry, Fear, Disgust, when an input is given it should apply the above discussed preprocessing techniques to extract features then it should apply 2 methods on it i.e. keyword-based approach and POS Tagger approach and then Identify keywords that are the useful to the classifier from the input dataset. Then finally classify extracted features to their respective classes of emotion then Find and compare accuracy of the Emotions for both the methods.

Effort:7 hours

Extend dataset(epic 3)
We have an emotion based Keyword dataset and Phrase dataset. Key word dataset has key words which can be used for training the system by classifying keywords into different files according to their emotions. Phrase dataset has phrases which can be used for testing the system by classifying Phrases into different files according to their emotions or keep it as 1 big fat file These dataset may be not sufficient at high level, hence extend the these dataset u sing a tool called word2vec tool.

Effort:5 hours

Activity 3-GITHB Final Submission

Project title: Online banking system

High Level Requirements:

ID	Description
1	Create a user account
2	View the account details
3	Credit Money
4	Debit money
5	Transfer Money between accounts
6	Mobile number change: If user enters old mobile number it should display a message

Table 7:High level Requirement(Activity 3)

Low Level Requirements:

ID	Description
1	If a user is below 18 year old, he should not be permitted to create account
2	It should take password from user and match it with the user database
3	If user credits money which exceeds daily transaction time it should notify user, on successful credit it should display a confirmation mail
4	If user tries to debit money from the empty /low balance account It should pop up an error
5	It should check whether both accounts are valid

Table 8:low level requirement(activity 3)

Design:

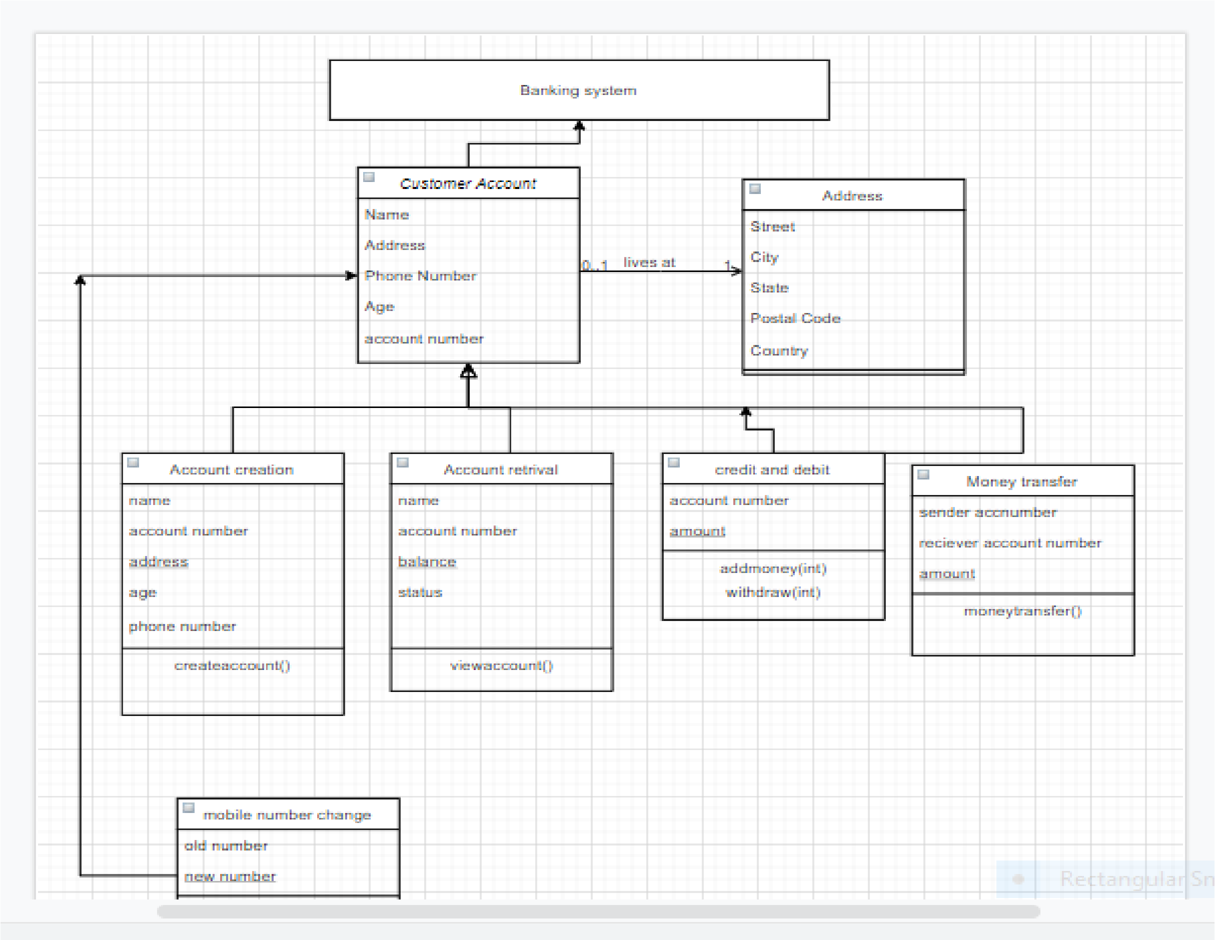


Figure 9:Class diagram for banking system

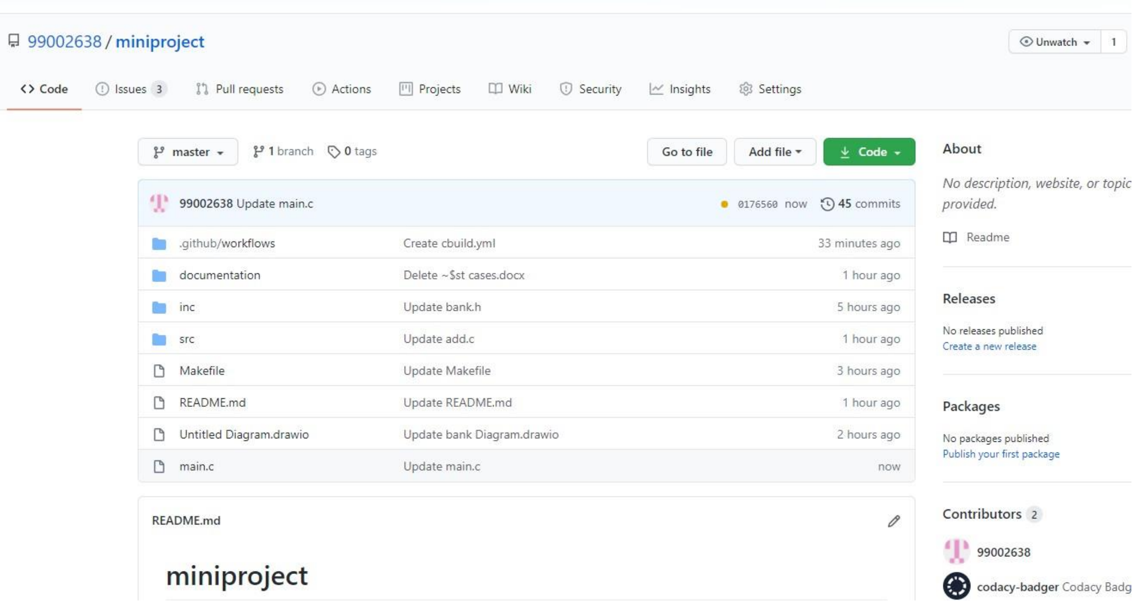
Test cases:

ID	Description	Pre condition	Expected input	Expected output	Actual output
1	Create a user account	If a user is below 18 year old, he should not be permitted to create account	User below 18 year old Ex age < 18	Account cannot be created	Account cannot be created
2	View the account details	It should take password from user and match it with the user database	User enters password with allowed inputs	If password matches: display the details If password does not matches: display the error	If password matches: display the details If password does not matches: display the error
3	Credit Money	If user credits money which exceeds daily transaction time it should notify user ,on successful credit it should display a confirmation mail	User enters credit amount based on pre condition	On valid transaction: successfully credit On invalid transaction: display error message	On valid transaction: successfully credit On invalid transaction: display error message
4	Debit money	If user tries to debit money from the empty /low balance account It should pop up an error	The sport riot caused too much anger in players and fans	On valid transaction: successfully debited On invalid transaction: Invalid transaction	On valid transaction: successfully debited On invalid transaction: Invalid transaction

5	Transfer Money between accounts	It should check whether both accounts are valid	User enters 2 account numbers and the amount to be transferred	On valid transaction: successfully transferred On invalid transaction: In valid Transfer	On valid transaction: successfully transferred On invalid transaction: Invalid Transfer
6	Mobile number change	If user enters old mobile number it should display a message	User enters new mobile number	On valid input: Mobile number changed On invalid in valid: this is the old number	On valid input: Mobile number changed On invalid in valid: this is the old number

Table 9:Test cases (activity 3)

CI Workflow:



99002638 / miniproject

99002638 Update main.c · 0176560 now · 45 commits

.github/workflows	Create cbuild.yml	33 minutes ago
documentation	Delete ~$st cases.docx	1 hour ago
inc	Update bank.h	5 hours ago
src	Update add.c	1 hour ago
Makefile	Update Makefile	3 hours ago
README.md	Update README.md	1 hour ago
Untitled Diagram.drawio	Update bank Diagram.drawio	2 hours ago
main.c	Update main.c	now

README.md

miniproject

About

No description, website, or topic provided.

Readme

Releases

No releases published
Create a new release

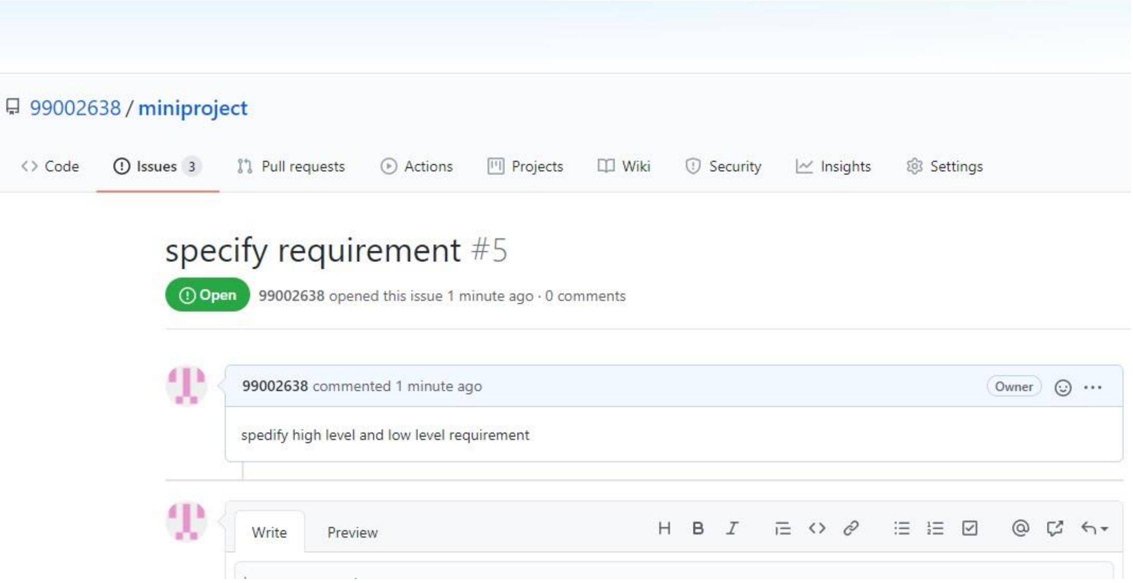
Packages

No packages published
Publish your first package

Contributors 2

99002638
codacy-badger Codacy Badg

Figure 10:git commits



99002638 / miniproject

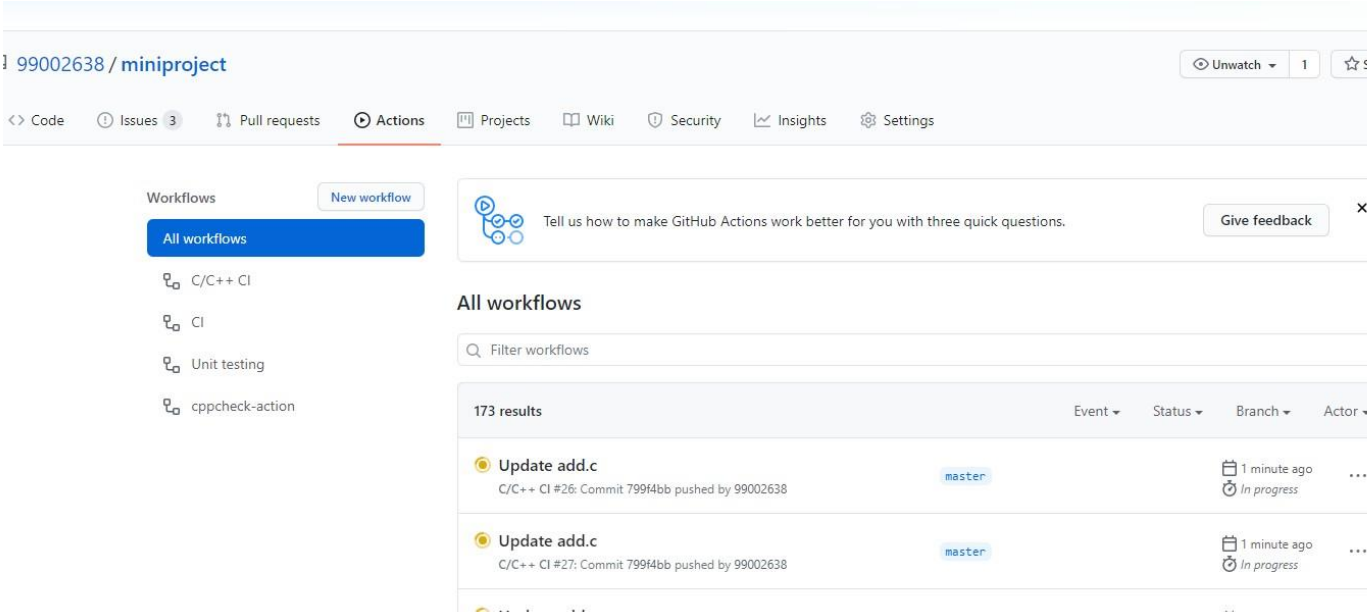
specify requirement #5

Open 99002638 opened this issue 1 minute ago · 0 comments

99002638 commented 1 minute ago

spedify high level and low level requirement

Figure 11: Issue created



Tell us how to make GitHub Actions work better for you with three quick questions.

All workflows

173 results

Update add.c
C/C++ CI #26: Commit 799f4bb pushed by 99002638
master
1 minute ago
In progress

Update add.c
C/C++ CI #27: Commit 799f4bb pushed by 99002638
master
1 minute ago
In progress

Figure 12:Git Workflow

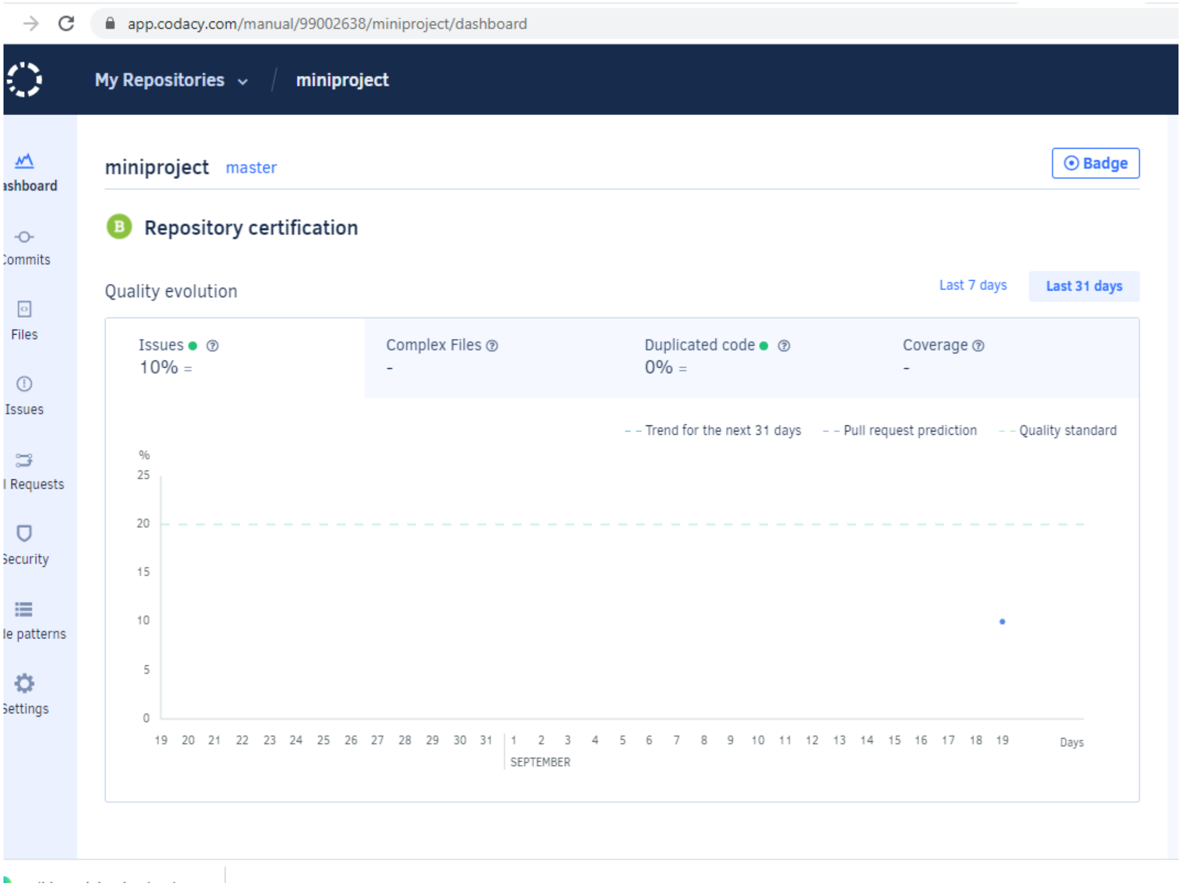
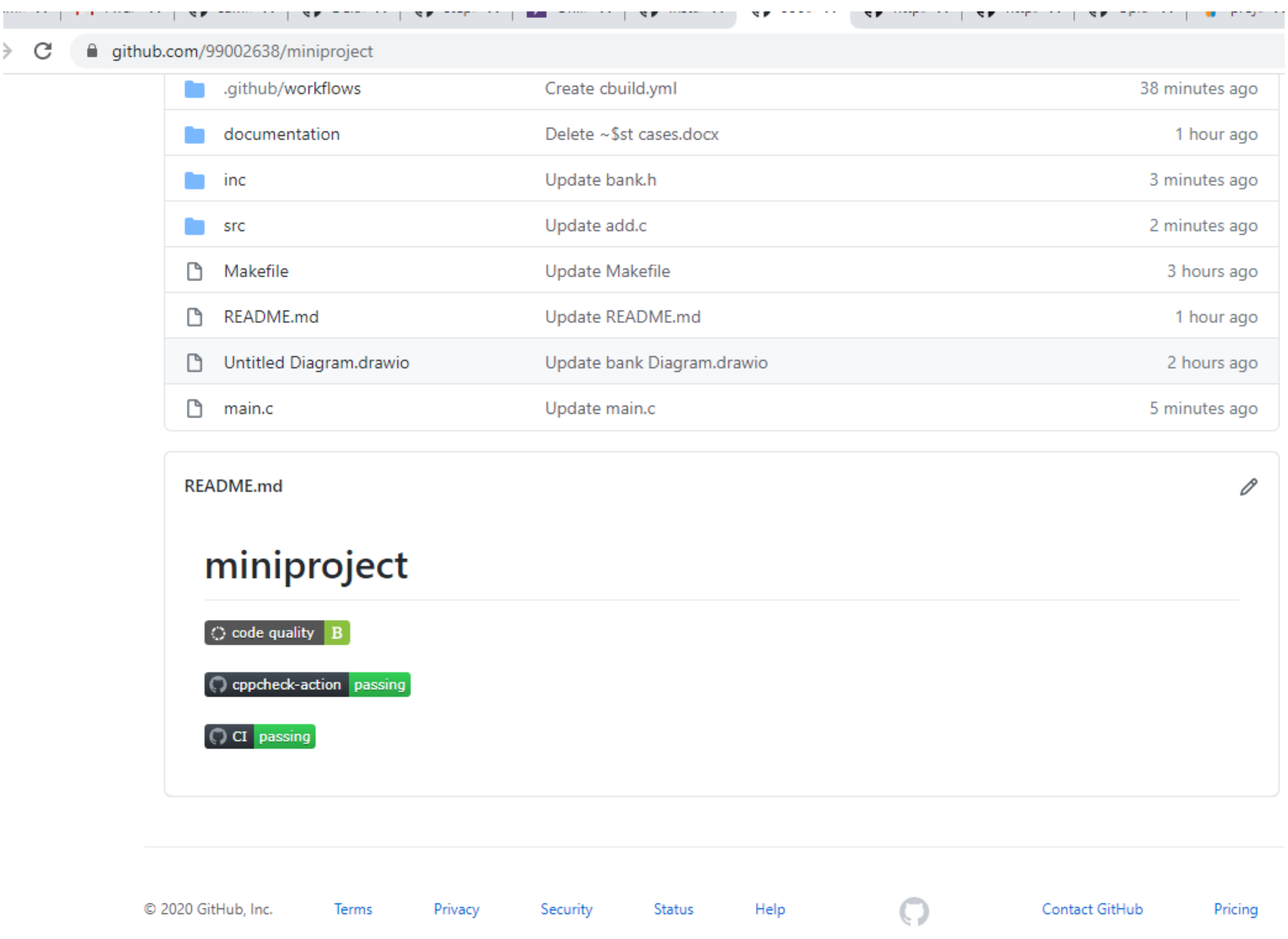


Figure 13: Code Quality



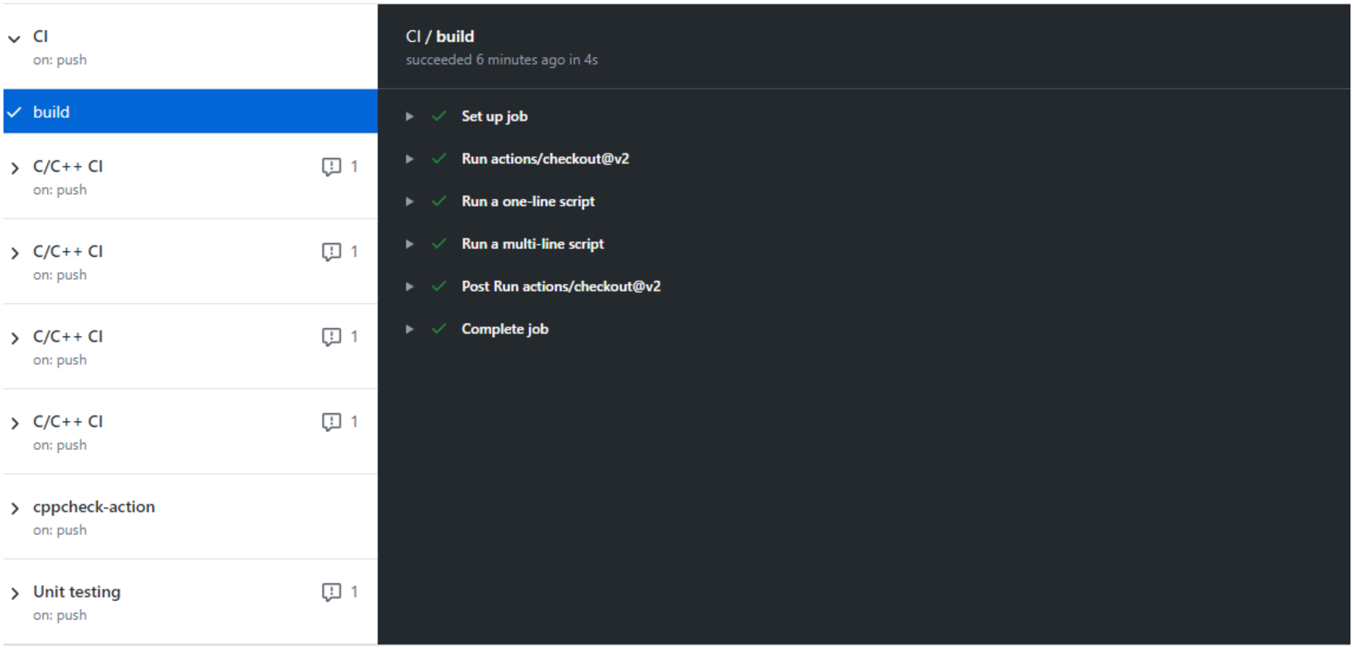
github.com/99002638/miniproject

.github/workflows	Create cbuild.yml	38 minutes ago
documentation	Delete ~$st cases.docx	1 hour ago
inc	Update bank.h	3 minutes ago
src	Update add.c	2 minutes ago
Makefile	Update Makefile	3 hours ago
README.md	Update README.md	1 hour ago
Untitled Diagram.drawio	Update bank Diagram.drawio	2 hours ago
main.c	Update main.c	5 minutes ago

README.md

miniproject

Figure 14: Badges



CI
on: push

✓ build

C/C++ CI 1
on: push

C/C++ CI 1
on: push

C/C++ CI 1
on: push

C/C++ CI 1
on: push

cppcheck-action
on: push

Unit testing 1
on: push

CI / build
succeeded 6 minutes ago in 4s

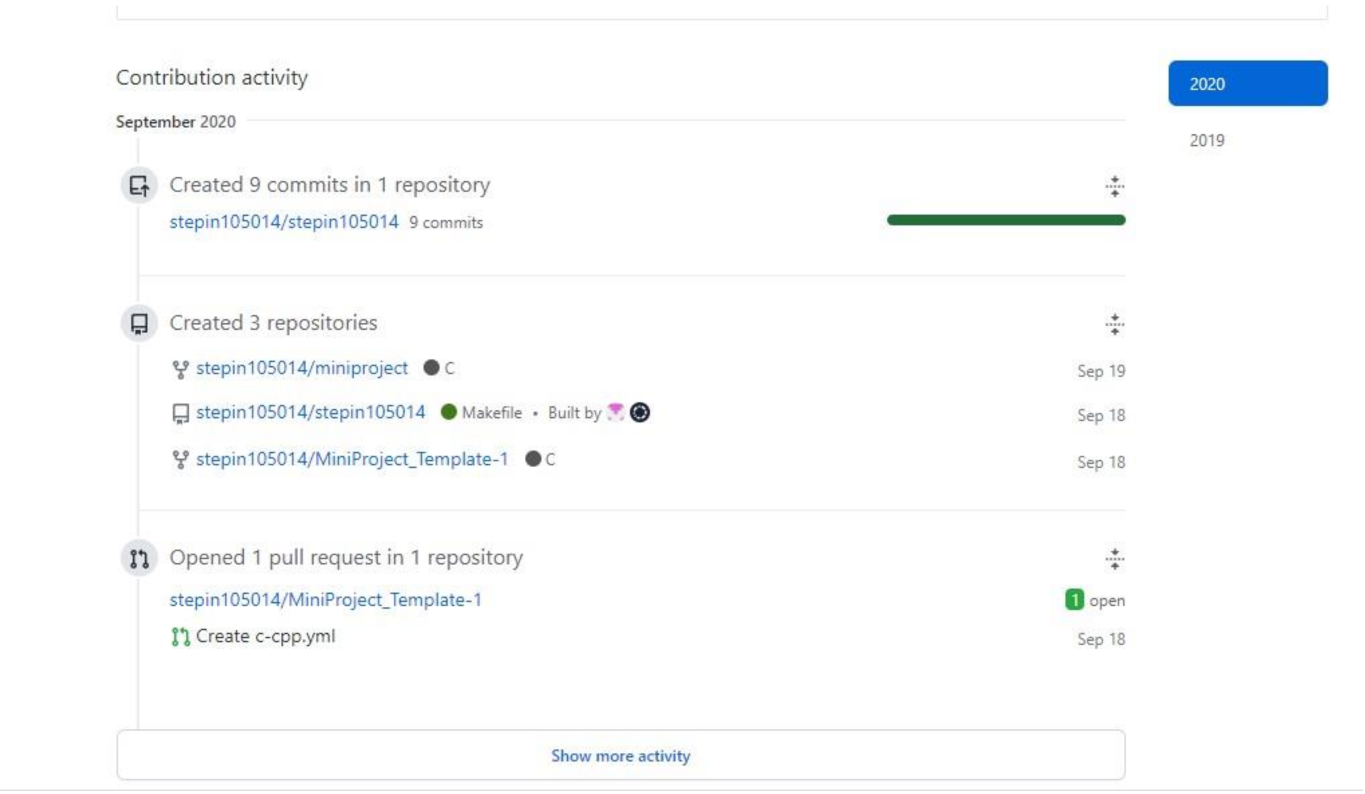
- ✓ Set up job
- ✓ Run actions/checkout@v2
- ✓ Run a one-line script
- ✓ Run a multi-line script
- ✓ Post Run actions/checkout@v2
- ✓ Complete job

Figure 15:Build

Link To github Repository: https://github.com/99002638/miniproject

Appendix

Completion of 5 basic github courses



Contribution activity

2020

2019

September 2020

Created 9 commits in 1 repository

stepin105014/stepin105014 9 commits

Created 3 repositories

stepin105014/miniproject ● C Sep 19

stepin105014/stepin105014 ● Makefile • Built by Sep 18

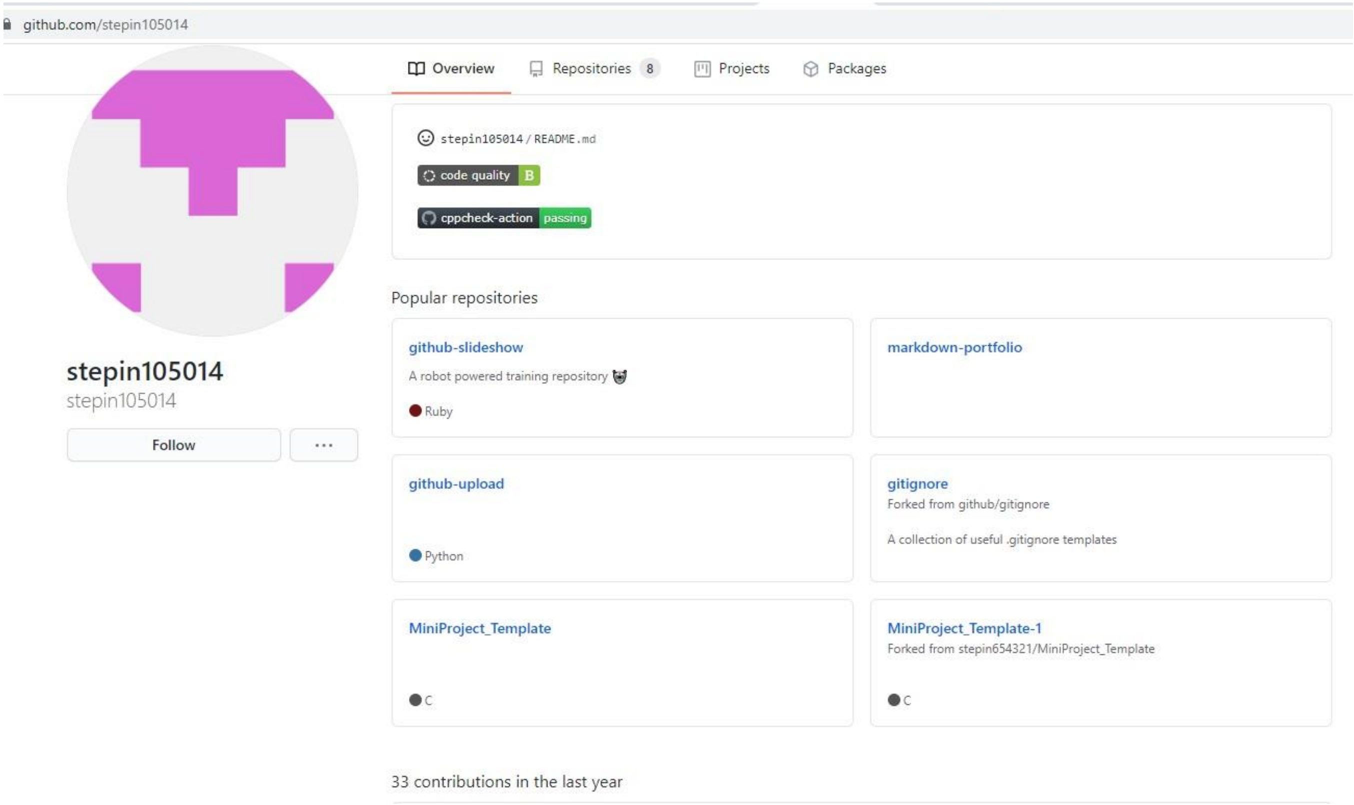
stepin105014/MiniProject_Template-1 ● C Sep 18

Opened 1 pull request in 1 repository

stepin105014/MiniProject_Template-1 1 open

Create c-cpp.yml Sep 18

Figure 16: Github basics



stepin105014 / README.md

code quality B

cppcheck-action passing

stepin105014
stepin105014

Popular repositories

github-slideshow
A robot powered training repository
● Ruby

markdown-portfolio

github-upload
● Python

gitignore
Forked from github/gitignore
A collection of useful .gitignore templates

MiniProject_Template
● C

MiniProject_Template-1
Forked from stepin654321/MiniProject_Template
● C

33 contributions in the last year

Figure 17: Repositories(after completing those tasks)

Link to the Stepin_Repository: https://github.com/stepin105014

References:

[1]. Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." Ain Shams Engineering Journal (2014).

[2] https://www.researchgate.net/publication/225045375_Emotion_Detection_from_Text

[3]https://www.tutorialspoint.com/uml/uml_activity_diagram.ht

ml

[4] [4]https://en.m.wikipedia.org/wiki/Use_case_diagram

[5] https://en.m.wikipedia.org/wiki/Data-flow_diagram

[6] https://en.m.wikipedia.org/wiki/Sequence_diagram

[7] https://www.javatpoint.com/online-banking-project