Project pitch

RepoRangers

Team

Name	Student number	Email
Ewa Kaleta	2745573	e.k.kaleta@student.vu.nl
Naomi Maronic	2740042	n.maronic@student.vu.nl
Vincent Kohm	2726735	v.n.kohm@student.vu.nl
Zohaib Zaheer	2735075	z.zaheer@student.vu.nl

Overview

GitHubMiner is a system that allows the user to clone and analyze a GitHub repository by providing the URL of the repo. The system has a command-line user interface. The user can type in various commands to extract information about the repository. The user can also quit/restart the system, which will result in deleting the currently analyzed repository. Additionally, it is also possible to generate a report with the information extracted by the system.

The primary type of users we target are developers, programmers and overall Github users - mostly people familiar with programming as the design (CLI) is easier for them. Our other main stakeholders are development teams, project managers and investors.

In short, the process goes as follows:

- The user provides the repo URL (private repos only available after authentication)
- Repo is cloned, and the user can track the progress of the cloning.
- The user can extract various information about the repo by using appropriate commands (repo statistics, sorted information (rankings), list of commits, etc.).
- The user can quit (to close the app completely) or restart (to go back to the starting screen). Those commands will also trigger the action to delete the repo from the system.

Functional features - using MoSCow prioritization

ID	Short name	Description	Priority	Champion
F1	Setup	Users can add a GitHub repository to the system by providing the URL of the repo.	**Must have -** This is an essential feature. Users must be able to add their repository, so they can extract information about it.	Ewa
F2	Cloning progress	Users should be able to track the progress of the cloning of the repository during the setup.	**Should have** - Users should be able to see how long the cloning will take but it is not essential for the system to work.	Ewa
F3	User interface	Users must be able to perform different commands using a command-line interface. The information should be presented to the user in text format.	**Must have -** Users must be able to use the system via a CLI (this design choice is important, because our stakeholders should be familiar with this UI)	Ewa
F4	Data parsing	Translating git log commands into data structures and formatting the output in the CLI.	**Must have** - Users must be able to view the requested data in a prettier format than git log returns.	Zohaib
F5	Reset/Quit	After quitting/restarting the system the repository should be deleted from the system. If app is restarted users should be able to go back to the starting screen to add and analyze a new repo.	**Must have -** Users must be able to analyze a new repository after restarting and start with a “clean” system if they quit and re-open the app.	Zohaib

Functional features

ID	Short name	Description	Priority	Champion

F6	Commands	The user can mine different information from the github repo by issuing command-line instructions following this syntax: `command + [target]* + [arguments]*`, where the (*) indicates that the parameter is optional.
		
		The available commands are the following:
		• `ranking + [commits, contributors]*`
		  ○ commits arguments - `[churn, recent]*`
		  ○ contributors arguments - `[commits, weekend, time, weekdays]*`
		• `stats + [commits, contributors, files]*`
		• `commits`
		• `history`
		• `help`
		• `restart`
		• `quit`
			Must have - this is the core feature of the system that will allow the extraction of information from a repo.	Naomi + Vincent

F7	Optional commands	Not specifying any targets/arguments in commands, where they are marked as optional, results in showing all possible information from the command. Ex. `stats` will show all statistics (commits, contributors and files), `ranking commits` will show all commits rankings (churn and recent) . This makes it easier for the user if they want to extract more than one information.	**Should have -** this feature makes it easier for the users to extract multiple information at once	Naomi

Functional (bonus) features

ID	Short name	Description	Priority	Champion
B1	Report (BONUS)	Users can generate a report of chosen metrics by typing the command `[report]`. The report will be generated as a new txt file.	**Should have** - users might want to generate a report to save the extracted information in an external file.	Vincent
B2	Choose report info (BONUS)	Users are able to choose what they want to include in the report. The information that can be included in the report are those provided by the commands `[ranking, stats, history, commits]` By default all information extracted by the system will be included in the report.	**Could have** - users might want to only include specific information in the report	Vincent
B3	Authentication (BONUS)	Users can add their private repositories to the system by logging into their GitHub account first and authorizing the system. After a successful log-in a link to private repo can be provided, otherwise only public repo can be accessed.	**Could have** - users might want to access private repos, because many project are not publicly available	Zohaib

Quality requirements

ID	Short name	Quality attribute	Description
QR1	Instantaneous results	Responsiveness	The system shall take no more than **500 ms** to display the metrics chosen by the user after the repository has been cloned.
QR2	Independence of commands	Maintainability	The system shall consist of commands that are **independent** of each other and thus enable the addition and deletion of singular commands.
QR3	Notification	Usability	The system shall notify a user and ask for a **correction**, in case a user enters an invalid URL.
QR4	Portability	Usability	The system shall **run on all major operating systems**, namely linux, windows and macOS.

Time log

Our team *RepoRangers* is planning to have weekly meetings on Thursday 1330h to discuss our user stories. During this meeting we will assign different tasks to different group members. This approach is inspired by the *Scrum methodology*

Team number		61		
Member	**Activity**		**Week number**	**Hours**
Zohaib	Prelimirary feasability research		1	3
Ewa	Added functional requirements and prototype design of the system		1	3
Naomi	Define functional and quality features and checking the ISO standards and defining stakeholders		1	3
Vincent	Created a collaboration space for the team and added quality requirements and defined stakeholders		1	3
			TOTAL	12

Signed contract

SD Team Contract