

Software Requirements Engineering
[SE-2001]
Project Deliverable - 2
Submitted To: Dr. Wafa Basit

Project name:

AsaanLang

Team Name:

Dream League V

Team Members:

- 1) Eehab Saadat (22L-7944)
- 2) Fayyez Farrukh (22L-7987)
- 3) Uffan Mehmood Khan (22L-7947)
- 4) Zain Iqbal (22L-7897)
- 5) Muhammad Ahmad Karim Qureshi (22L-7857)

Table of Contents:

Content	Page No.
1. Project introduction 1.1. Abstract 1.2. Scope statement	1-1
2. Requirement Elicitation 2.1. Techniques & Outcomes 2.2. Constraints	1-6
3. Requirements Classification	6-8
4. Feasibility Study	9
5. Diagrammatic Representation 5.1. Context Diagram 5.2. 0-Level DFD	10
6. Persona	11
7. Scenario	12
8. References	12

1. Project Introduction

1.1. Abstract

The AsaanLang project represents a groundbreaking initiative aimed at removing language barriers in the world of programming. Designed to empower the vast community of native Urdu and Hindi speakers, AsaanLang introduces a programming language with a simplified Roman Urdu/Hindi syntax, revolutionizing the way coding is approached. By bridging the linguistic gap, AsaanLang seeks to create a more inclusive software development environment, making it accessible to both novice and experienced programmers who may lack native-level English proficiency. We aim to ensure the provision of equal opportunity of learning to code for people who spent their lifetimes through Urdu/Hindi medium educational institutes and were denied equal access. Our vision is to foster diversity within the software industry while providing a sense of pride and belonging to South Asian software engineers, ultimately transforming the landscape of programming for the masses.

The following document presents a crucial step in the software development process, serving as a Requirements Elicitation document. It lays the foundation for understanding and capturing the essential needs, objectives, and constraints of the project at hand. By meticulously gathering and documenting the requirements through various requirements elicitation techniques such as brainstorming, domain analysis, interviews and questionnaires, we aim to create a blueprint that identifies and classifies project stakeholders, requirements, user personas and scenarios ensuring a clear and comprehensive understanding of the project's scope and objectives. This document is instrumental in guiding the project team towards successful planning, design, and implementation, ultimately leading to the realization of a solution that meets the defined needs and objectives.

1.2. Scope Statement

The scope of AssanLang is to create a user-friendly language learning platform tailored for professionals having Urdu/Hindi as their primary language and struggling to learn, adapt and catch-up to the rapidly changing technological landscape.

2. Requirements Elicitation

2.1. Techniques & Outcomes

2.1.1. Brainstorming

Multiple physical and virtual, both formal and casual, meeting sessions of varying 15-45 minutes of duration were conducted at pre-decided times. Major key points were noted such as UI-design for the web based interface, preliminary language syntax, transpiler execution model, module division and workflow was decided and noted down on paper or recorded as voice recordings for future reference and as a skeleton structure for preliminary project and business requirements.

2.1.2. Domain Analysis

A plethora of open-source projects, research articles and literature were reviewed and analyzed. The reviewed open-source projects, available on *GitHub*, namely UrduScript and BhaiLang, were analyzed and compared to evaluate the similarities and differences of the features and practicality (see section 4 for a tabulated summary of the findings). It was primarily identified that similar pre-existing projects' ability to stay market

relevant and practical was due to their nature as casually-developed hobby projects with no formal practical goal, lack of maintenance and discontinued developer support in contrast to the rapidly changing landscape of supporting technologies. Furthermore through ample literature review, both the product development and product market practicality was estimated as such as how the *Washington Post* ranked Urdu/Hindi to be the second most spoken language with 588 million speakers world wide, in contrast to English being placed at third with a total of 527 million.

Additionally, a survey conducted by *Gallup Pakistan* in 2023 demonstrated that an astonishing 53% participants receive(d) education in Urdu-medium educational institutions. Adding to this a comparison between compiler, interpreters and transpilers was conducted through further literature review from multiple renowned sources such as *StackOverflow*, *DEV Community*, *GeeksForGeeks*, *Medium* and *Scaler* which helped decide the project's preferred choice to be the transpiler execution model due to a plethora of reasons which include ensuring and balancing maximum optimization with feasible developmental complexity within the scope of available resources and also the removal of double abstraction to achieve the execution time of the target language.

2.1.3. Interviews (Response Rate = 50%)

Two extremely valuable and domain-skilled resource persons, Mr.Nafees and Mr.Sheryar, were reached out to and were interviewed in a virtual call of duration 30-45 minutes. Both interviewees were self-taught programmers who started their programming career early in their teenage years and are currently pursuing Bachelors in Computer Science from National University of Computer and Emerging Sciences, Lahore. They both are founding developers of their own currently maintained custom open-source programming languages namely *qscript* and *plutonium*, both of which are interpreted domain-specific languages (DSLs).

These interviews were structured with pre-decided questions and incorporated open questioning to incorporate the diversity and subjectivity of the responses. Both the interviewees demonstrated an agreement on the lack of software engineering principles, especially in the planning and requirements engineering phase, as a consequence of decreased code changeability and modularity, to be the major contributors to project bottlenecking and rise in redo effort and time. Both interviewees, when asked about what they would do if they were to hypothetically redo their project from scratch, responded and emphasized greatly on properly structured planning and requirements engineering, while focusing on early-stage formal syntactical definition in Backus-Naur Form (BNF) notation or other syntax definition standards to avoid developmental complexity, scope creep and bugs down the development which should be further tested pre-development through parser generators.

Both interviewees, when asked to comment on our project's practicality and market audience, responded positively especially for its potential impact in the education sector by making learning algorithm design accessible in urdu/hindi-medium educational institutes or native audiences and focused that its true value would be ensured if the proposed language, avoiding feature bloating, provided an extended ecosystem support of the target language, incorporating popularly used, if not all, modules. Both focused on a linear development transpiler developmental style and suggested some valuable

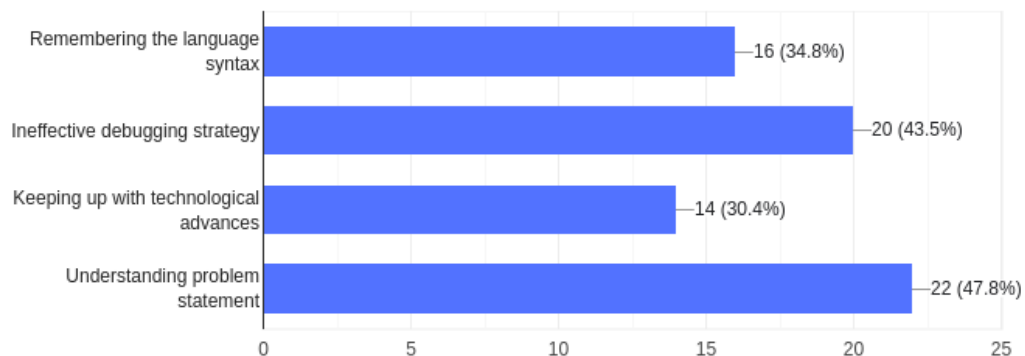
syntax definition guidelines such as addition of syntactic sugar and the principle of don't repeat yourself (DRY) as quoted by Dennis Ritchie the founder of C language, to avoid syntactic redundancy and ensure syntactic simplicity instead. Both agreed that a transpiler would be the best option for balancing language development and performance as ensured by the removal of double abstraction as observed in interpreted languages.

2.1.4. Questionnaires (Response Rate = 47%)

A questionnaire consisting primarily of closed questions but also open-ended questions to accommodate subjective suggestions, such as for feature addition and recommendations, was designed to collect and survey audience feedback and project practicality. This questionnaire was designed with the goal of gathering project requirements and gauging public stance regarding our project idea of making programming accessible to native Urdu/Hindi speakers. A total of 46 responses were collected of whom 56.5% were Computer Science students, 23.9% were software engineers, 6.5% self-taught programmers and 13% others. Out of these, 63% respondents had been programming for 1-3 years, 23.9% for less than a year, and 13% were not a programmer at all. 78.3% of the respondents ranked their English proficiency within a 4-5 range while the rest 21.7% ranked it to be amid 1-3 on a 1-5 ranking scale.

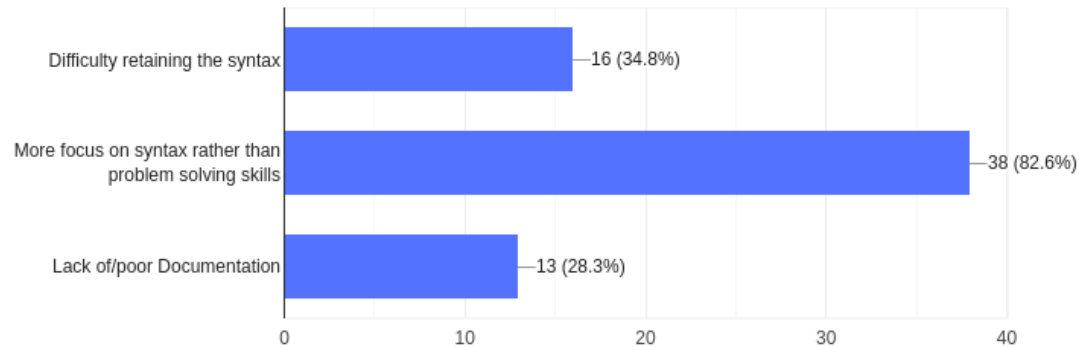
A summary of the project-relevant key-metrics in conjunction with their respectively traced inferred and extracted requirements are listed below:

- 47.8% respondents listed understanding the problem statement, 43.5% listed it to be ineffective debugging strategies, 34.8% listed syntax retention in memory, and 30.4% listed keeping up with technological advances as their major day-to-day challenges as a coder. The high response rate on the former three metrics demonstrates the significant impact of a programming language's syntax on its users, which in conjunction to available research^[2], shows how syntax written in native language could help understand and retain the syntax better and hence consequent to better debugging strategies with enhanced code comprehension and readability, catering to aforementioned user preferences.

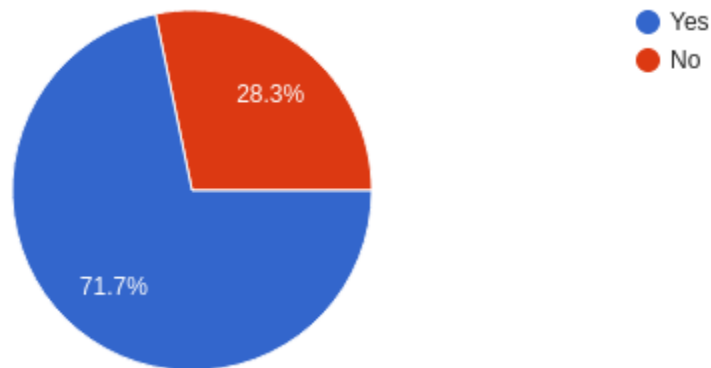


- 82.6% listed more focus on syntax than on problem solving skills while 34.8% selected difficulty in syntax retention and 28.3% opted for lack of/poor documentation as the biggest challenge for new programmers learning to code. These collected metrics greatly helped us to understand the potential obstructions a learning programmer might face and hence support the decision of

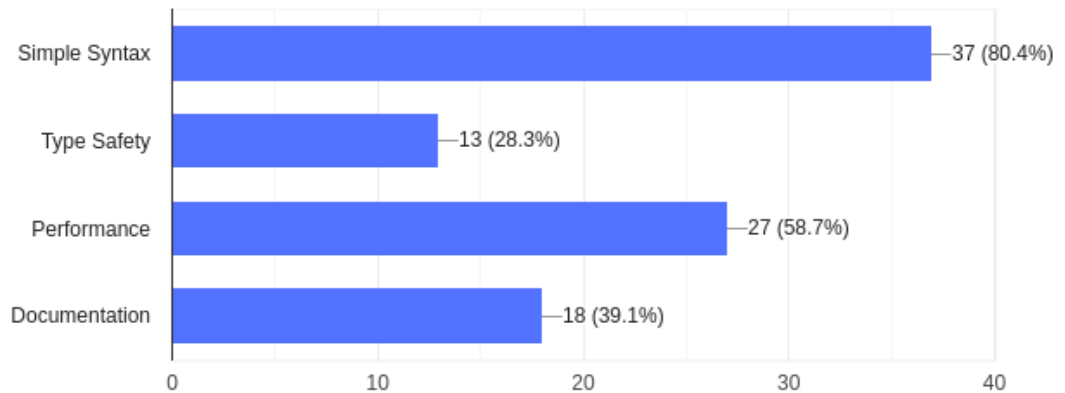
developing a native-language based programming language, as per research^[1], coding in such a language helps enhance a person's problem solving skills significantly as it reduces the comprehension and retention barrier between the language and the user, while the addition of well documented and illustrated code would help lower the learning curve for struggling new native programmers.



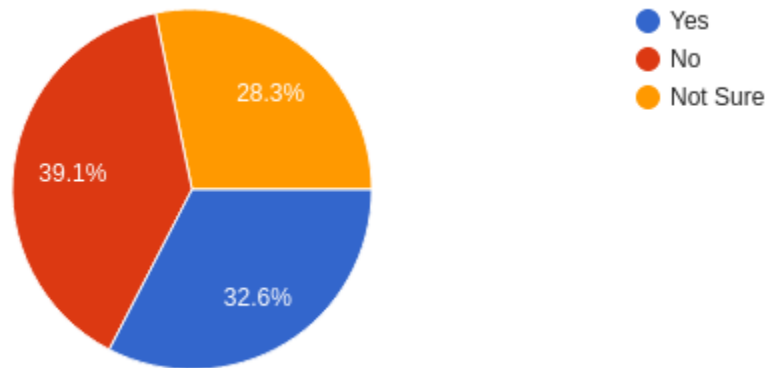
- 71.7% answered 'Yes' and 28.3% answered 'No' as a response to finding pre-existing online code playgrounds and executors such as OnlineGDB, CodePad and W3schools to be useful for learning and using a programming language. This traces back to our provision of a web-based playground editor that would aid its users to transpile and run code in the source language, while also providing the functionality to view target language transpiled code for enhanced learning and backend transparency.



- 80.4% of the respondents chose syntax simplicity, user-friendliness and understandability as the most crucial quality attribute for the development of a new hypothetical native Urdu/Hindi language-based programming language while 58.7% emphasized on the language performance and compilation speed, 28.3% on type safety and robustness and 39.1% on documentation. It can be evidently seen here that a language developed in native language with maximum syntactic sugar ensuring clarity and readability is the most looked forward to feature. The choice of code transpilation helps ensure the other two attributes by removal of the double abstraction layer and hence achieving both the equivalent type safety and performance speed of python. Furthermore, detailed documentation with illustrations and examples would also help to lower the learning curve further.



- 32.6% of respondents voted 'Yes', 28.3% voted 'Not Sure' and 39.1% voted 'No' when asked if the given preliminary syntax of our language would enhance their experience as a coder. These metrics helped analyze the market trends and interests and cater to whether the demand and the practicality of the project.



2.2. Constraints

2.2.1. Scope Constraints

Latinized Urdu Script is preferred over Arabic Urdu Script due to technical advantages, as tokenization in Arabic Urdu would be more complex and time-consuming. A transpiled language interpretation has been chosen rather than one with a compiler or interpreter, and the addition of advanced features such as object-oriented programming, complex data structures and some built-in functionality has been left out to maintain feasibility within our current limitations.

2.2.2. Time Constraints

Rapid development may not permit the full realization of all potential features and functionalities. Complex tasks such as object-oriented programming and certain built-in functionality may require more time than available, leading to trade-offs and prioritization of critical components.

2.2.3. Technical Constraints

Certain advanced features may pose technical challenges due to the learning curve associated with their implementation or dependencies on external technologies. While we aim for a robust language, these constraints may necessitate focusing on foundational elements initially and gradually expanding capabilities in future iterations as the team acquires the required expertise.

3. Requirements Classification

Business Requirements

- **B.R.1.** To enable urdu/hindi speakers to code without the prerequisite in english language proficiency.
- **B.R.2.** Engage a community of programmers and developers in the advancement of non-English syntax based programming languages

External Interface Requirements

- **E.I.R.1.** A main web-interface/playground is to be developed for code editing and execution.
- **E.I.R.2.** A Command-Line-Interface will be developed for interaction, code writing, transpilation and code execution.
- **E.I.R.3.** AsaanLang transpiler must be compatible with python interpreter for successful execution of the source code.

Functional Requirements

- **F.T.1**
 - a. AsaanLang shall support declaration and manipulation of primitive data-types including integers, floating-point numbers, strings, and booleans and python-built-in data types (list, tuple, set and dictionary).
 - b. It shall provide support for assignment operations.
 - c. It shall support a comprehensive set of arithmetic functions, including addition, subtraction, multiplication, division, modulus, and exponentiation along with both unary and binary arithmetic operations.
 - d. It shall allow implementation of standard conditional statements.
 - e. It shall allow implementation of code loops.
 - f. It shall allow users to create user-defined functions and standard function calls, supporting return value and passing arguments and recursion.
- **F.T.2**
 - a. The project shall include a source-to-source transpiler with python as target language.
 - b. AsaanLang syntax must include implementation or mapping of key python expressions and built-in python data structures.
 - c. The project shall implement an error-handling mechanism to detect and report syntax errors in the source code.
 - d. The project shall include a symbol table to store the language grammar and identifier information.

- **F.T.3**
 - a. AsaanLang transpiler's web-interface shall be responsive to allow for use on differently sized screens.
 - b. The web-interface shall have an input box that allows users to write code and an output console to display the results of executed code.
- **F.T.4**
 - a. Static web pages that provide documentation, guides, and other informative content shall be created similar to python's official documentation.
 - b. The system shall provide installation and setup guides to assist users in installing any required software components, dependencies, or libraries.
- **F.T.5**
 - a. The project shall provide a comprehensive mapping of Python libraries and modules in AsaanLang
 - b. The project shall be able to generate .py files directly from AsaanLang source code without the need for code execution.

Non-Functional Requirements

- **F.T.1**
 - a. AsaanLang shall be dynamically typed.
 - b. AsaanLang shall support run-time memory allocation and deallocation and garbage collection.
- **F.T.2**
 - a. The project shall include a Lexer for lexical analysis and token identification.
 - b. The project shall include a tokenizer to tokenize the source code with reference from symbol table, label and generate Keyword Object Model.
 - c. The project shall employ a parser based on Backus-Naur Form (BNF) for grammar checking, to validate the syntactic correctness of the source code against the defined grammar rules and generate an Abstract Syntax tree (AST).
 - d. The project shall have a code generator component to convert AST into code in the target language.
- **F.T.3**
 - a. AsaanLang transpiler shall be integrate with command-line interface to develop and run AsaanLang scripts.
- **F.T.4**
 - a. The project shall maintain comprehensive documentation including details on language syntax, semantics, standard libraries, and best practices.
 - b. The project shall foster and encourage an active and engaged user community.

Key Quality Attributes

- **Q.A.1. Maintainability:** Asaanlang transpiler must be modular and allow for adaptation to new changes in the python development environment (updates in python, python libraries and frameworks) to keep AsaanLang relevant for active developers.
- **Q.A.2. Robustness:** Code execution of AsaanLang ensures that the system can handle errors, unexpected input variations, and adverse conditions while continuing to function correctly.
- **Q.A.3. Readability:** The source code should be readable incorporated with syntactic sugar (simplistic, clear, concise and easy-to-follow) and minimally redundant following the don't repeat yourself (DRY) syntax design principle. The source code should be readable by users of all Urdu/Hindi proficiency levels.

System Requirements

- **S.R.1.** The system utilizes a python interpreter to execute the transpiler generated target python code and render its output.

User Requirements

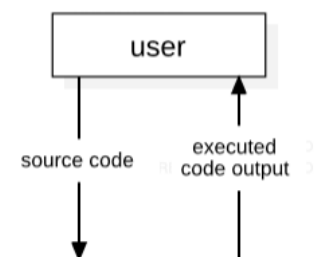
- **U.R.1.** Develop robust, general purpose pro operable command-line programs on logical, executable code.
- **U.R.2.** Learn the fundamentals of programming and software development

4. Feasibility Study

From domain analysis (2.1.2) and interviews (2.1.3), we conclude that; AsaanLang transpiler should be limited to basic functionalities initially (see 2.2.4 for elaboration). There is a considerable scope for AsaanLang to be used as a tool to learn programming for novice urdu/hindi speaking coding, as AsaanLang shall target top issues faced by those learning to code, like retaining the syntax (34.1% of respondents), improving problem solving skills ((Bernardo, 2002), 82% of respondents) and poor documentation (27.3% of respondents). However, for an industry standard programming language, continual maintenance, extensive language ecosystem and an active support community are prerequisites which will be challenging, resource-intensive and time taking to develop and should be put aside for now. Domain analysis showed little to no room for reusability of existing components from similar projects. A summary of similar projects and their major features in context of AsaanLang is given below:

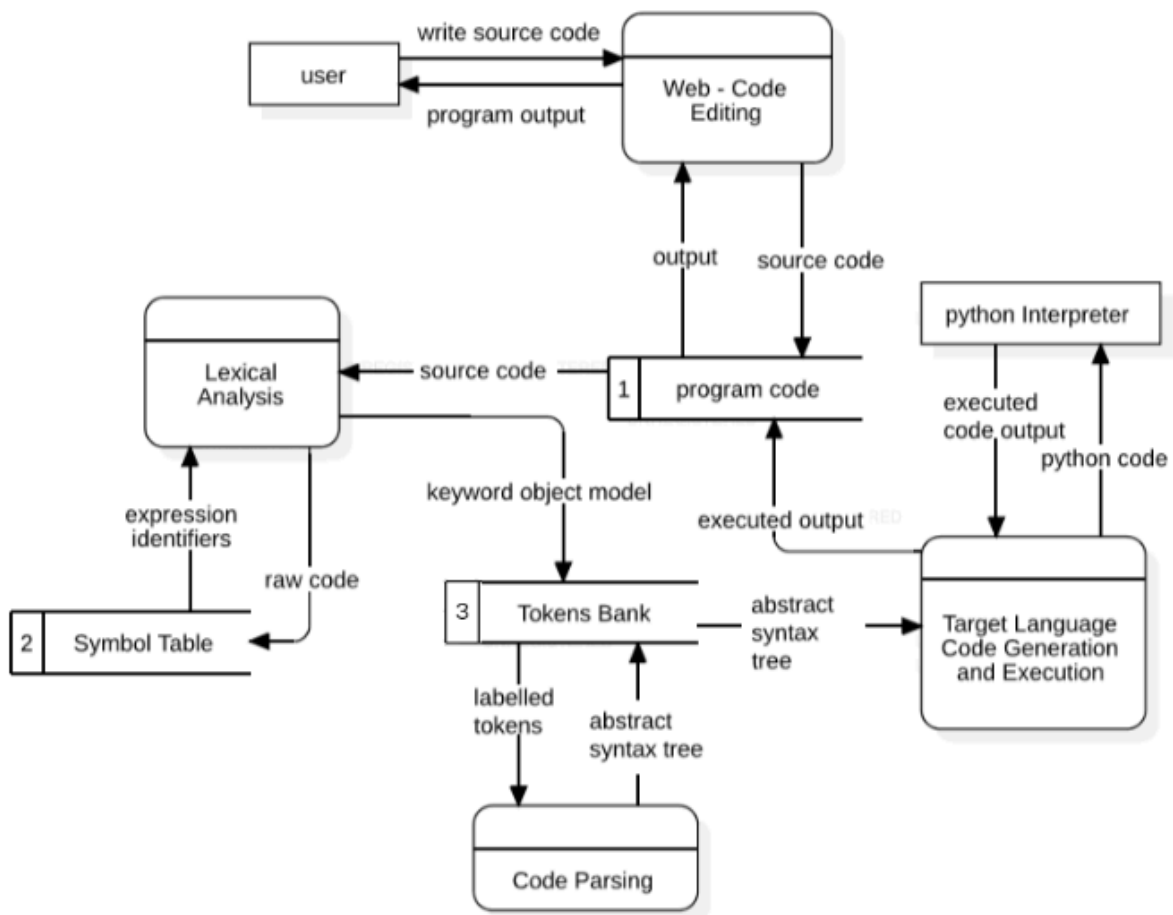
Programming Language	Base/Target Language	Professionally Acceptable	Type Safety	Dynamically Typed	Web Interface	Documen-tation	Urdu/Hindi syntax
TypeScript	Javascript	Yes	Yes	Yes	No	Yes	No
Nim	C/C++	Yes	Yes	No	No	Yes	No
BhaiLang	Typescript	No	Yes	Yes	No	No	Yes
UrduScript	Javascript	Yes	No	Yes	Yes	Yes	Yes
DogeLang	Python	No	Yes	Yes	No	No	No
AsaanLang	Python	Yes	Yes	Yes	Yes	Yes	Yes

5. Diagrammatic Representation



5.1 Context Diagram:

5.2. 0-Level Data Flow Diagram:



6. Persona

Major user classes, learners and industrial enthusiasts, identified through interviews and questionnaires can be ideally represented through following two personas:

6.1. Primary Persona

Haris Rauf is a 15 year old urdu and punjabi speaking individual belonging to a lower median income household based in Bahawalpur. Haris has access to an android smartphone but no computer literacy or usage experience. He has recently completed his matric from a local urdu-medium school and nowadays has a lot of free time at disposal. He has exceptional analytical skills and often thinks out of the box. Despite having a strong aptitude for growth and self-learning, his English proficiency and vocabulary are subpar owing to his unimpressive educational background. Studying basics of programming in WB BASIC in matric sparked his interest in programming and computer sciences in general but his knowledge from PTB's computer science book is redundant and outdated. Haris has decided to learn to code in his free time. However, he finds his unfamiliarity of English language and inaccessibility to a pc/laptop as significant challenges.

Goals:

Haris needs a simple programming language which is easy to understand for urdu speakers, comes with easy documentation and one that can be edited and executed using a responsive web-interface that works on mobile devices.

6.1. Secondary Persona

Rohit Sharma is a 26 year old electrical engineer, graduated from Sri Shakthi Institute of Engineering and Technology, Coimbatore. He uses Hindi as his native language and only has functional English language proficiency. He is an inexperienced self-taught programmer who recently switched his career from junior electrical engineering to a junior software developer intern in a small software house, 2 months ago. His company mostly works in python. Rohit never learned algorithm design in a formal manner and observed that his problem solving skills are lagging behind his colleagues. He finds it difficult to retain the syntax and is also not comfortable with bloating in python syntax.

Goals:

Rohit wishes for a non-bloated programming language which allows for better cognitive processing (simple to process for his mind) and has a more memorable syntax. Rohit must be able to convert this code directly into python code using command line so that it can be integrated with other modules created by the rest of his team. In addition, such a language must have support for common python libraries that he uses on a regular basis.

7.Scenario

The programmer opens the AsaanLang web interface. They are presented with the code editor and a 'Getting Started' button. They click on the getting started button. They are then redirected to another page which has a tutorial and a code editor. They watch the tutorial which provides a step-by-step guide on how to write a simple program, and implement that code in the code editor. Suppose the programmer makes a syntactical mistake in the code and presses the 'run' button. The system detects the mistake and displays an error message and asks the programmer to correct the syntax and press 'run' again. Otherwise, if there was no syntax error, the source code is passed to the transpiler and executed as python code by python interpreter. The code output is then displayed on the output console of the code editor and systems return to standby state.

8. References

1. Bernardo, A. B. I. (2002). Language and mathematical problem solving among bilinguals. *The Journal of Psychology*, 136(3), 283–297.
<https://doi.org/10.1080/00223980209604156>
2. Gregory, J., & Varley, S. (2004). Cross-linguistic transfer in bilingual aphasia: Evidence from a case of pathological language mixing. *Aphasiology*, 18(3), 213-231.
<https://doi.org/10.1080/09571730385200181>
3. The Karachiite. (2015, September 18). Urdu/Hindi: The second most spoken language in the world.
<https://thekarachiite.com/culture/urduhindi-the-second-most-spoken-language-in-the-world.html>
4. Gallup Pakistan. (2021, March 21). 59% Pakistanis say they can speak Urdu fluently; 18% say they can speak English fluently: Gallup Survey. <https://gallup.com.pk/>
5. QScript Scripting Language: <https://github.com/Nafees10/qscript>
6. Plutonium Interpreted Language: <https://github.com/shehryar49/plutonium-lang>
7. Questionnaire Google Form Link: <https://forms.gle/zN6XXNEf6PFojqZ8>