International Journal of Engineering and Management Research e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
www.ijemr.net https://doi.org/10.31033/ijemr.12.5.68

Developing a Cost-Efficient Employee and Project Management System for Small-Scale Software Development Startups

M.R.A. Perera[1], S.P. Rupasinghe[2], E.M.S.D. Ekanayake[3], H.M.S.Y. Nanayakkara[4], D. I. De Silva[5] and S.Vidhanaarachchi[6]
[1]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[2]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[3]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[4]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[5]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[6]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA

[1]Corresponding Author: ravinduanjanaperera@gmail.com

ABSTRACT

Startups are a vital and significant part of the software development landscape. They are the first to implement and embrace new technologies and changes in existing technologies. Even though the vast majority of startups are minor in scale, they represent a significant market share. They develop innovative products within a limited time frame with less capital. Their need to work within a limited time frame guides most startups to embrace the agile methodology. These companies also work with proprietary data and procedures essential to maintain competitiveness with other companies in the market. Employee management systems can reduce the workload on human resource management, and project management systems can increase the efficiency of software development. Both these systems are essential tools for a startup. Even though many tools provide these functionalities in the market, they were not developed specifically for startups. This research aims to create a single service optimized for a small employee base that can provide both of these functionalities in the most cost-efficient and secure way.

Keywords-- Startup, Agile, Employee Management System, Project Management System

I. INTRODUCTION

Software development startups (SDS) are new businesses focused on developing and monetizing new software products as their main income stream. Ries [1] defines a startup as a human institution designed to create a new product/service under conditions of extreme uncertainty. Blank's [2] definition of a startup is a temporary organization that seeks a scalable, repeatable, and profitable business model to grow. Both of these definitions

Highlight that a startup is a highly unstable institution that operates under a thin margin of error. The main reason for this is that startups face challenges, such as developing cutting-edge products, acquiring paying customers, and building entrepreneurial teams [3].

In addition to the challenges faced by other startups, since SDS "develops software under highly uncertain conditions, tackling fast-growing markets under a severe lack of resources"[4], they have to use their limited resources efficiently to develop their products rapidly. Due to this reason, most startups use agile development methodology as opposed to traditional software development lifecycles. The agile method's

Focus on people over processes and iterative development are particularly suited to startups that work with rapidly changing user requirements and typically have smaller development teams with less complex organizational hierarchies. Even though startups are relatively unstable organizations they fill a valuable and vital role in the software development landscape. For example, Startups apply existing knowledge to open up unexpected avenues for improvement [5]

SDS will work on multiple different projects within the same time frame. A single project may be handled by one or more development teams depending on the complexity of the project and organization hierarchy. One employee may be assigned to multiple projects due to the scarcity of talent in startups. Projects are divided into multiple steps that must be completed and employees have different target requirements they must fulfill. Documentation related to every project must also be stored securely. Managing these requirements using manual methods or multiple different tools will increase the operational cost of the SDS. Using a dedicated software solution optimized to manage projects can cut down on this cost and increase the efficiency of software development on the SDS. According to research made on applying agile methodologies in developing a costing system using agile project management leads to a higher acceptance of the final product by clients [6]. This and the reasons mentioned in the above paragraphs are why agile project management methods are the main focus while creating the project management part of this work.

SDS will typically have a limited number of employees usually ranging from 10 -100. Similarly, to other businesses, SDS also has to manage employee details, record

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
https://doi.org/10.31033/ijemr.12.5.68

employee attendance, payroll management, and other related functions. Due to the lack of resources, using a dedicated software solution to manage the employees instead of investing more resources in accounting and human resource personnel is more cost-effective for SDS.

For SDS investing capital to obtain and maintain two different software solutions to manage employees and projects can be an unsustainable drain on resources. For a small-scale company with a limited number of employees and a less complex organizational hierarchy, most of the dedicated employee management systems (EMS) or project management systems (PMS) on the market offer irrelevant functionalities with a higher cost. A system specifically designed for startups to be cost-efficient and offer core functionalities of EMS and PMS on a single software service will be useful for many SDS.

The rest of this work is structured in this way. The Literature Review would discuss similar products on the market and related research. In the Methodology section, the system's architecture and design will be discussed. In the Discussion, overall analysis and steps for future implementations will be discussed.

II. LITERATURE REVIEW

Attendance marking is a core function in EMS. Akshat Gupta et al [7] have researched to develop an employee management system for the hotel industry. The main focus of the research paper is implementing a biometric-based attendance marking system to reduce user error and the overhead cost of human resource management. Even though this work highlights the superiority of using an automated attendance system for employees, it does not present how the system is to be implemented.N. Kar et al's [8] research on a system designed for marking attendance in a classroom describes implementing a facial recognition system using "the eigenface approach for face recognition which was introduced by Kirby and Sirovich in 1988 at Brown University."[8]

Kanban charts are a lean method used in agile development.P. S. M. dos Santos et al's [9] worked on a synthesis study using 20 selected primary studies to provide "an empirically-grounded reference for decision-making in practice."[9]. Out of the sixteen benefits they searched for, the benefits with the most confidence identified from the kanban approach are 1) work visibility,2) control of project activities and tasks,3) flow of work 4) time-to-market [9]. Out of these 4 benefits, time to market is especially important for SDS. The 'organizational culture' is the major challenge identified with the kanban approach. SDS would have a lower effect from this challenge when compared to established companies with more organized work culture.

Jira[10] and Azure board[11] are two of the most popular project management systems for software engineers in the world in 2022. Both of these products are Software as a Service(SaaS) products that provide their functionalities as cloud-based products. Even though both of these products provide the ability to manage projects following agile methodology they have been specifically developed to provide these functionalities and the cost of the service can run up depending on the employee numbers and hierarchy in the SDS

Open HRMS[12] and Zoho people[13] are examples of popular employee management systems in the market. open HRMS is an open-sourced human resource(HR) management system that has a business model that charges companies by the service package they chose. Zoho people is a primarily cloud-based service that charges companies based on the number of users. Even though both of these options provide a multitude of options for companies their focus on creating the system is more established companies with the typical corporate structures. They would not provide an SDS with the same value as a more dedicated HR management system that was developed just for SDS.

III. METHODOLOGY

A. Tools and Technologies

In this section, the tools, and technologies used to develop the proposed system would be discussed. Java Script (JS) is the language used in the development of the system. The specific libraries and frameworks used will be discussed in more detail in the proposed system section.

The software was developed following the agile methodology. Microsoft azure board was used for project management. Automated testing was done using Selenium. Sonarqube was used for code validation and better code quality. git was used as the version control system and GitHub was used as the remote repository.

B. Proposed system

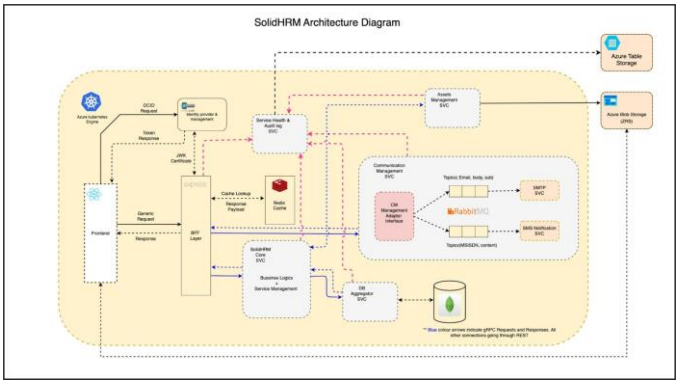

Figure 1: System architecture

As mentioned above in the introduction section startups work with proprietary data and services in a highly

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
https://doi.org/10.31033/ijemr.12.5.68

competitive environment. Corruption of data or third-party accessing this data without the company's knowledge can lead to massive losses to the SDS and can sometimes lead directly to bankruptcy. The ability to store and manage these data without the intervention of a third party can be important for SDS. The best option for deployment from a data security and integrity perspective is an on-premise deployment of the software in a server center managed by the company. But this option is costly for SDS. The cost of maintaining an on-premise server center combined with the cost of personnel and systems need to provide the required cyber security means that this would require a higher initial investment. The lack of resources SDS faces means this option is not the best choice for the implementation of the service. A cloud-based implementation of this service will be more economical. Even though there are concerns about a third-party having access to the data this can be mitigated to a certain extent by creating a separate instance for every SDS that uses the service. Due to these reasons, it was decided to design the system to be deployed in Azure Kubernetes Service (AKS)[14].

React [15] is a declarative, component-based JavaScript library focused on creating interactive user interfaces. React is also designed to be able to be used in any technology stack. For these reasons React was chosen to implement the Frontend of the system

Authentication and identification of the users are major requirements in the proposed system. Ensuring only the employees of the company can access the data stored on the system according to their authority levels is very important to SDS due to their need for data security. The Identity providers and management service are implemented using Keycloak identity providers[16]which is an "open source identity and access management solution"[16].

Frontend calls Backend for Frontend(BFF) microservice using REST API as Generic Request response mechanism. BFF is a service that calls API endpoints of other services so that the front end cannot access other microservice's API endpoints directly[17]. This increases the security of the system and ensures that the system can be easily adapted to different types of clients in the future if needed. For example, if a mobile application or biometric device needs to be connected to the system in the future BFF layer ensures that business logic is separated from the frontend client type and it can be easily adapted to the different requirements of the frontend.

The request time of the backend is a fairly important metric in the development of software. High latency in response time and the backend server not responding can ruin the smooth user operation of the software. When a large number of users use this service through a longer time scale even the difference of milliseconds can be a considerable loss of efficiency. Redis[17] is an in-memory data structure store, used as a distributed, in-memory key-value database, cache, and message broker. Redis is used to reduce the request-response time of the backend. The repetitive calls are cached so that computational cost is decreased and the latency of the response from the backend is reduced.

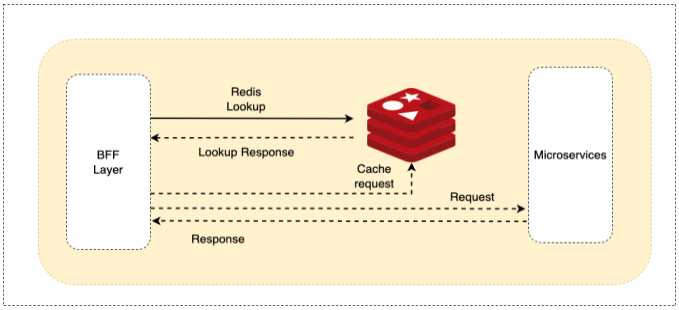

Figure 2: Redis communication

Fig.2 represents the communication between the BFF layer and Redis. All the names mentioned inside the brackets in the following paragraph refer to the names mentioned in fig.2 .The Redis is implemented between the BFF layer and other microservices BFF layer first sends a request(Redis Lookup) to check whether the required data is already cached. Then Redis sends responses (Lookup Response) to the BFF layer. If the data was already cached BFF layer sends the data to Frontend. If not BFF layer sends a request(Request) to the relevant microservice and waits for a response (Response). After receiving the response BFF layer sends the relevant data to Frontend and sends a request(Cache Request) to Redis asking it to cache the data.

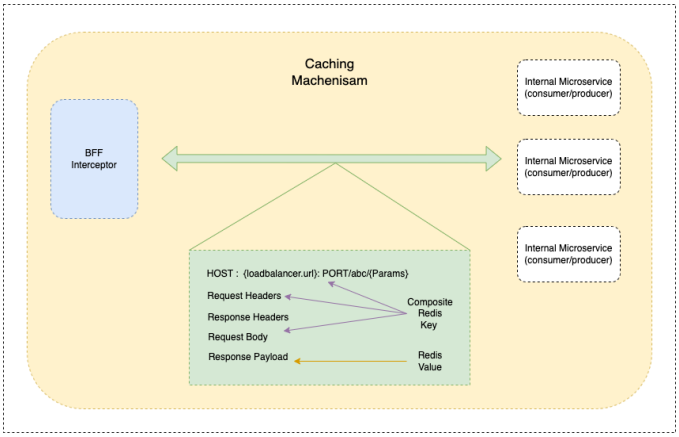

Figure 3: Caching mechenisum

Fig 3 shows how a cache record is stored in the system. The BFF interceptor is the part of the BFF layer that calls Redis. The header and Body of the Request and Header of the Response are stored as the key value of Redis and the Response Payload is stored

Solid HRM core service is a microservice used to handle the business logic of this software. Calculations and business logic that needs to be implemented in this system will be handled through this service. Separating business logic into a separate microservice improves the maintainability and readability of the code base. The different API endpoints for different functions that need to be implemented will also be implemented here. The BFF layer will identify and call the relevant API endpoint to the function requested from the Frontend. This service will also decide what database operations need to be executed. Then it will call DB Aggregator service.

DB Aggregator service is used to handle database operations. Database security is an important part of a secure system. Project and employee data stored in the system by SDS are confidential and with a high need for data integrity. By limiting the number of services with access to the database we can improve the security of the database. It's the only service with direct database access. Other services should communicate through the DB Aggregator service to perform database operations.DB Aggregator service is only used for DB operations. Deciding the logic of which DB operations must be executed is handled through the Solid HRM core service as mentioned in the above paragraph. The BFF layer shouldn't directly call the DB aggregator service under any circumstance. This ensures that no outside party can access the DB aggregator service directly.

A database is used to store the data relevant to the system. MongoDB[18] is the database used to develop the system.MongoDB is a cross-platform,document-oriented database program. It is a non-relational database program. It uses documents and collections to store the data. Documents take a JSON-like format. MongoDB is highly customizable when compared to a relational database. The ability to easily add, edit or remove fields on a document when compared to the difficulty in changing columns in a table in the relational database means MongoDB is the superior choice for this system. Since many different SDS will use this system and the data fields they store of employees and projects will be unique to each SDS MongoDB is the better choice.MongoDB is also a highly scalable database. Since data in MongoDB is not tightly coupled as in a relational database we can horizontally scale MongoDB database by distributing data in multiple data nodes. MongoDB automatically scales databases by using Sharding operations.

Storing images and raw data in a database can lead to huge database costs and bottlenecks. Using a separate cloud object storage solution to store images and other raw files relevant to the system can remove these disadvantages. Azure blob storage[19] is a cloud-native, scalable, and secure object storage solution. Assets management microservice is used to access the Azure blob storage.

Ensuring the visibility of user actions in each service and getting an overall view of all services across the system is important to ensure better system security. The audits give administrators of the system the bailey to track the behaviors and operations which happened in the system in case of a possible security breach. Health checks asses things services need to execute their functionalities. These could be database connections, dependencies, System properties, etc. Storing these service health reports can help diagnose potential breakdowns or unexpected behaviors in the system. Service health and audit log service are used to monitor user activities and collect service logs. service logs are stored in the Azure table storage[20]. Azure table storage is a NoSQL key-value storage.

Communication management microservice is used to manage to send email and SMS notifications functions. Rabbit MQ is an open-source message broker. RabbitMQ produces a queue to handle multiple simultaneous requests to communication management microservice. Then SMTP or SMS notification service consumes the relevant queue

Two different methods are used to implement API calls in the system. "Representational state transfer (REST) is a software architectural style that describes a uniform interface between physically separate components, often across the Internet in a client-server architecture"[21]. gRPC[22] is a high-performance remote procedure call framework.gRpc has a low latency and a high-speed throughput communication which makes it ideal for microservices where message efficiency is paramount. Rest can use custom payloads and widespread support of third-party tools and it is the de facto standard in the industry which makes it ideal for microservices opened to external parties. In fig 2. blue lines are used to represent gRPC API calls.red lines represent Audit log requests which are REST API calls. The rest of the requests and responses represented in black also use REST API calls.

C. Equations

The attendance management system of the work stores the check-in and checkout time of the employees. Equation (1) is used to calculate the monthly salary of an employee.

$$S = \left(\frac{m}{160}\right) \times \left[\left(\sum_{n} \frac{(c-i)}{3600}\right) + 8(h+l)\right] \quad (1)$$

S= salary for the selected month
m= monthly salary
c= outgoing Unix time for that day
i= incoming Unix time for that day
h = number of holidays in the month
l = the number of leaves in the month
n= number of days in the month

IV. DISCUSSIONS

After implementing the product 2 main opportunities for further development were identified.

Manually marking attendance was identified as a tedious and error-prone approach. Users may forget to sign in or sign out from the system, they may make fraudulent claims of attendance and it takes a considerable amount of time for the attendance marking procedure. Implementing biometric recognition for automated attendance marking is a major step to reducing overhead costs and user error and increasing the efficiency of EMS[7]. Azure cognitive services can be used to implement a facial recognition system. Since the system is deployed in AKS using azure services is easier for the management of the system. It's also more cost-efficient than developing a custom facial recognition service.

Registering new employees to the system, and asking them to remember login credentials for another system was also identified as a potential pain point. Giving users the ability to sign up for the system instead of HR management departments posed a security risk of not being able to identify which are the actual employees of the SDS. For the SDS that use corporate emails using them as a way to verify employees is an answer to the problem. keycloak provider offers the Oauth2 social integration capability. Implementing a social login function to this system can be easily done using this provided functionality.

So employees can access this system using Google cloud identity federated to the corporate email.

REFERENCES

[1] E. Ries. 92014). *The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses*. New York: Crown Business.
[2] S. Blank. (2020). *Startup Owner's Manual: the step-by-step guide for building a great company*. John Wiley.
[3] C. Giardino, S. S. Bajwa, X. Wang & P. Abrahamsson. 92015). Key challenges in early-stage software startups. *Lecture Notes in Business Information Processing*, pp. 52–63. DOI: 10.1007/978-3-319-18612-2_5.
[4] Paternoster et al. (2014). Software development in startup companies: A systematic mapping study. *Information and Software Technology 56*(10), 1200-1218. DOI: 10.1016/j.infsof.2014.04.014.
[5] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek & P. Abrahamsson. (2016). Software development in startup companies: The greenfield startup model. *Transactions on Software Engineering, 42*(6), 585–604.
[6] V. Jiménez, P. Afonso & G. Fernandes. (2020). Using agile project management in the design and implementation of activity-based costing systems. *Sustainability, 12*(24), 10352. DOI: 10.3390/su122410352.
[7] Akshat Gupta, Anisha Kundu & Rik Das. 92019). Automated attendance system for efficient employee management: a biometry based approach. *International Journal on Recent Trends in Business and Tourism, 3*(3), 117-121.
[8] N. Kar, M. K. Debbarma, A. Saha & D. R. Pal. (2012). Study of implementing automated attendance system using face recognition technique. *International Journal of Computer and Communication Engineering*, 100–103. DOI: 10.7763/ijcce.2012.v1.28.
[9] P. S. M. dos Santos, A. C. Beltrão, B. P. de Souza & G. H. Travassos. (2018). On the benefits and challenges of using kanban in software engineering: a structured synthesis study. *Journal of Software Engineering Research and Development, 6*(1). DOI: 10.1186/s40411-018-0057-1.
[10] https://www.atlassian.com/software/jira.
[11] Azure Boards | Microsoft Azure," azure.microsoft.com. https://azure.microsoft.com/en-us/products/devops/boards/.
[12] C. Technologies. "Open HRMS | MOST ADVANCED OPEN SOURCE HR MANAGEMENT SOFTWARE," Open HRMS. https://www.openhrms.com/.
[13] "HR Software Solutions | Cloud based HRMS | HR System | Zoho People," Zoho. https://www.zoho.com/people/.
[14] "Managed Kubernetes Service (AKS) | Microsoft Azure," azure.microsoft.com. https://azure.microsoft.com/en-us/products/kubernetes-service/ (accessed Oct. 27, 2022).
[15] Facebook, "React – A JavaScript library for building user interfaces," Reactjs.org, 2022. https://reactjs.org/.
[16] "Keycloak," www.keycloak.org. https://www.keycloak.org/.
[17] Redis, "Redis," redis.io. https://redis.io/.
[18] MongoDB. (2019). The most popular database for modern apps. Available at: https://www.mongodb.com/.
[19]"Azure Blob Storage | Microsoft Azure," azure.microsoft.com. https://azure.microsoft.com/en-us/products/storage/blobs/.
[20] "Table storage | Microsoft Azure," azure.microsoft.com. https://azure.microsoft.com/en-us/products/storage/tables/ (accessed Oct. 27, 2022).
[21] Wikipedia Contributors, "Representational state transfer," Wikipedia, Jul. 24, 2019. https://en.wikipedia.org/wiki/Representational_state_transfer.
[22] "gRPC," gRPC. https://grpc.io/.