Digitized system for TinkerBell Garden

Software Engineering Project 5
Team 24 - TeamID 509

27th April 2022

Nguyen Thu Hieu - hieu.nt194761@sis.hust.edu.vn
Nguyen Khanh Linh - linh.nk194789@sis.hust.edu.vn
Bui Tran Hai Quan - quan.bth194821@sis.hust.edu.vn
Vu Dinh Minh - minh.vd194804@sis.hust.edu.vn
Nguyen Minh Tuan - tuan.nm194876@sis.hust.edu.vn
Nguyen Truong Giang - giang.nt194751@sis.hust.edu.vn
Le Duc Huy - huy.ld194777@sis.hust.edu.vn
Mac Dinh Phu - phu.md194817@sis.hust.edu.vn

Hanoi University of Science and Technology

Contents

1 Executive Summary

The following proposed system is intended for the staff of Tinkerbell Garden, the administrator, and the customer visiting the playground. The staff and administrator of the park are represented by Professor Mai Anh Bui Thi - our primary client and Ms. Chau Nguyen Thi Minh who is very knowledgeable about the park's needs. The basic target of the development team is to implement a computer-based system to replace the manual management process on paper of the park. It will keep the basic functionality of the current paper version and add more advanced features to help the administrator of Tinkerbell Garden manage his customers efficiently, optimize the profit and reduce the workload and time of each staff member. The overall goal of the new system is to automate processes as much as possible, to bring better performance to the customers visiting the park, and attract more customers to the park. Therefore, the owner of the park can make effective management decisions to optimize the processing stage and maximize his profit.

2 Preliminary requirements analysis

2.1 Application Overview

Objectives

The main functionality of the system will be to manage the customers visiting Tinkerbell Garden in a fast and flexible way. The data of the customer then can be extracted to serve statistical purposes. In addition, it also provides the customer with a more intuitive and simplistic way to use lots of services in the park.

Business Objectives

The project aims to reduce costs, improve customer service, optimize the workforce, and store data more efficiently for future strategies. Through this digitized system, the project expects that more people will come to Tinkerbell Garden to experience the exciting games and enjoy the best customer service. It is also expected to greatly benefit the staff by reducing workload and increasing the ease of customer-related data management through a user-friendly interface.

A working prototype will be implemented and tested in time to be deployed at each milestone which is expected to be in May 2022, June 2022, and the final product is released lately in July 2022.

Current Business Process and Rules

Currently, every process in Tinkerbell Garden is done manually. The customers can buy tickets in the form of paper and the customers' detailed information is saved on spreadsheets in Excel software. Sometimes they want to organize special events for customers, the staff needs to send emails to each person one-by-one which takes lots of time and effort. The customers also need to interact directly with staff to buy the ticket, which is cumbersome. The new system hopes to solve the problems via a user-friendly interface such that everything can be done by a few mouse clicks automatically.

User Roles and Responsibilities

Administrator Role: The ability to sell tickets, view, edit and update data about customers, price, infrastructure, etc.

Customer's Role: They can receive notifications, book tickets on holiday and enter information about the park.

Interaction with Other Systems

The system will be built from scratch using standardized web packages or winform replacing an existing paper system. The team decided to go ahead with a web application due to the scalability and flexibility of the requirements.

Production Rollout Considerations

The system can be designed, developed, tested, and put into production in a phased manner as early as about three months. The staff can use this system after undergoing a short period of training. The customer can use the system to enjoy the services immediately easily because of the intuitive user interface experience.

2.2 Functional Requirements

Statement of Functionality

One kid can be accompanied by an extra parent (free) to buy tickets to the amusement park. Children's tickets have two types of price: price for playing every 2 hours and price for playing with no limit time within a day (until Tinkerbell Garden closes). When the child plays beyond the time limit, it will be charged 50k/30 minutes, and multiply it, parents must pay the excess amount before leaving the amusement park. The system must calculate time and price automatically for the customer at the checkout.

The system also shows ticket prices for some additional charged activities such as painting statues, painting pictures, playing video games,...

The system must provide the functionality to manage and make statistics of players' data each day. It must also manage the game activities and status of damage as well as the facilities of each game area.

The owner can hold special events on holidays to attract more people to come to the park. The customers can also register to be VIP members of Tinkerbell Garden to receive a lot of offers from the services and get discounts.

The system will be flexible and must be accessible from various devices, about 100 customers at any given time.

The system must store data of customers in the database to develop sound business strategies in the future.

The system will have different access levels so that different types of users may log in and out. It is also a responsive application to be viewed perfectly on both mobile and PC.

Security and User Capabilities

The software will support 2 types of users. Admin can access the system by logging in with their passwords. At that administrative login level, the staff will be given additional

permissions to access and edit the data of all customers. At the access level of the customer, they can only access some information about the park.

Non-functional requirements

The software will be installed and run on existing Windows systems. The system needs to function whenever a customer needs to access it. The users should be able to use the system effectively after one week at most. The performance must be fast and intuitive.

Optional Features

The system allows searching data of customers or generating data in comma-separated values to be imported into Excel.

The system may be developed on both winform and website systems. Only the website is part of the requirements.

The system may have a backdoor to backup data in an emergency case.

Usability

Usability issues such as speed of operation for the user interface, and the number of concurrency connections from both customers and staff will be important considerations.

Scope

The scope of the system includes handling lists of customers in several ways, including ticket buying and selling. The system also performs necessary calculations to produce intuitive data for statistical purposes and management of information about the visiting frequency and customer's tastes. The stored data are limited to the customers who visit Tinkerbell Garden and the infrastructure inside.

The system will not support critical changes to the fundamental way how information is stored. It also doesn't support changing the scope of the system. It will support changes in data management and the addition of new functionality.

3 Process To Be Followed

For this project, the team has decided to follow the Agile approach with the Scrum method which involves beginning with a user interface mockup and gradually adding functionality, changing the system throughout each sprint until the clients are satisfied with the final products. This method is chosen due to its advantage. The client can see the progress anytime and the running prototype can show to the client as soon as possible at the end of each sprint. The requirement is not clear and can be changed throughout the development process that is suitable for us to leverage the strengths of the Agile method compared to some traditional paper methods. If necessary and time permitting, redesign of the interface will occur based on client feedback. In each sprint, the team will be able to add the necessary software module to handle other functionalities.

Below is the proposed outline of each sprint and milestone including what our team expects to have completed at each stage

Process Outline

At each milestone, the team will present the most current version of the software to the clients for their testing and evaluation. Although the software will not initially be fully completed during the early presentations, it will give the clients an approximation of the functionality of the final product.

Note that the time mentioned below may not be the exact time of each release because the time here is limited to one semester and decided absolutely by the primary client.

3.1 1st sprint(26/5/2022) - Milestone 2 of the project

The first presentation

It may not be too specific but it normally demonstrates the current version of the system including provisional design. It will clearly show how the current version meets parts of the requirements. It also contains the progress graphs, timetable of the Scrum method, and descriptions of the whole process of making the application.

The first version of the application

The first version must meet parts of the client's specification which is the priority functionalities. It contains a user interface and some first functionalities must work properly. It is also tested carefully and can be released right away.

In detail, the first version of the application allows the admin to sign in and sell ticket. Therefore, customers can buy tickets at the counter of the amusement park. All information will be recorded in a security database. It also allows customers to view information about games and price of Tinkerbell Garden. It contains full source code from both backend and frontend.

3.2 2nd sprint(23/6) - Milestone 3 of the project

The second presentation

It may demonstrate how the current version of the system meets the client's requirements.

The second version of the application

The second version must allow the admin to manage customers visiting the amusement park. It will make statistics about customers each day. Admin can also keep track of the status of the infrastructure of each game area.

The system also shows ticket prices for some additional charged activities such as painting statues, painting pictures, playing video games,... It allows the admin to hold special events for discounts and send emails to customers to make announcements. Finally, it must also fix some bugs and improve the existing functionalities following the feedback of the clients in the previous sprint.

3.3 3rd sprint (28/7) - Milestone 4 of the project

The team plans to reserve one weeks before the final presentation so that the client can test the product in its intended environment with real users and data. All functional requirements will have been met before this point, any changes hereafter will only cover small details, such as aspects of the user interface.

The final documentation

It may demonstrate in detail every functionality of the projects in a formal way. It must contain system design, user interface design, and other information about the process of making this project. It also shows the client how to use the website and covers any desired and optional features that have also been implemented

The final version of the application

The final system will include all features that the team and the client have agreed are required. It must allow customers to register as VIP members and get more discounts. It is also set up some more components and tools to avoid security risks and improve load capacity. It must be tested carefully in every case and ready to be released as the final product.

4 Suggested deliverables

To satisfy the client's need for a digital solution to their current system, the following set of work products will be delivered to the client:

a) Periodic Presentations

Accompanying requirements, design, and final reports will be periodic presentations where the team will demonstrate different aspects of the software system in development. The team's Agile Scrum approach to software design means that different presentations may concentrate on specific areas of the client's need. These presentations are designed to give the client maximum understanding of how their needs are being addressed in the actual product. Feedback will be critical to these presentations, so that the team may use the client's comments to understand completely how the system meets the client's needs.

b) Website for TinkerBell Garden

This system will be the core deliverable for the client. It will consist of codes of both backend and frontend. The client has identified some major functionalities that this system will need to deliver:

+) Manage every transaction buying tickets. All data is stored to make statistics to show for the admin.

+) Customers can register to become VIP members and get discounts, book tickets and see the information online.

+) The administrator of Tinkerbell Garden can manage everything easily. They can hold special events to attract more kids to the playground.

c) Documentation for User

The client will be provided documentation both explaining how to use our system and describing its underlying structures. It contains almost every information about the projects.

5 Technical Feasibility

The feasibility of the technical requirements can be judged by identifying and outlining at least one technical method that will satisfy the client's needs. Since any technical solution must embody the client's requirements, it is useful to identify them:

1) The system can use data about all the customers visiting the amusement park to make meaningful statistics. It must also be easy for users to create, delete and update data quickly. This can be done by the fast query programming language SQL

2) The user interface must be intuitive and have a high performance. We can use React to make the frontend of the website, which can separate each part of the website into many different components and update only what we need. Users can make a lot of operations on a single webpage without having to wait for reloading. To make sure the customer satisfy with the user interface, we also use some design tools like Figma to create a read-only user interface and review it with client before coding.

3) The front end and database must interact with each other fast and efficiently. We can use NodeJS as a webserver to handle connections. NodeJS is an event-driven language which is very fast and can handle these problems.

4) The website must be able to handle about 100 connections at a time. We can use Nginx as a load balancer to handle many connections at a time and forward requests to the NodeJS server. It will increase the load capacity of the system. It is also a good way to prevent DDOS attacks.

5) Security needs to be maintained to ensure the integrity of the data, although no malicious or advantageous editing is expected. Password-protection and a login system (based on access-level or usertype) are sufficient.

Given this rough outline of requirements, a possible system design could be the system in Figure 1:

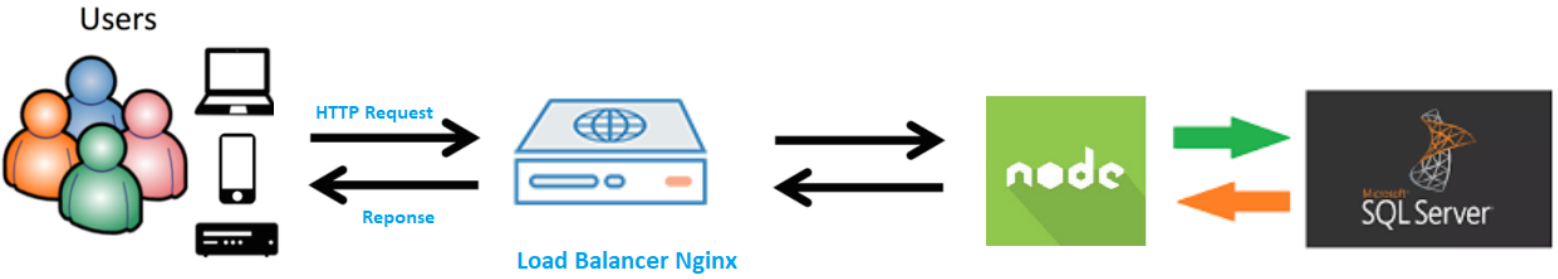


Figure 1. An abstract representation of the system design

In the above structure, SQL Server is a central database which stores all data about customers, VIP customers, infrastructure,... We can replace it by any type of relational database like MySQL,... The server side can be implemented by NodeJS which can be used by several other web servers like PHP, Java,... It will handle all the logic from requests

of users and provide an interface to interact with the database. Nginx is used as a load balancer to forward connection to the NodeJS server. It will increase the load capacity of the system by handling and dividing all connections properly.

In conclusion, there is at least one technically feasible solution to the proposed system. This feasible system would consist of a centralized SQL database, NodeJS server and a React web interface. The combination of these freely available software products and the team's own coding will satisfy the client's requirements. To further test the feasibility of this possible system, the team must consider that the client expects roughly 100 users to be able to simultaneously access the system. With the limited number of end-users, the hardware limitations of the library's central server should not be a problem.

Finally, it should be noted that the final system delivered to the client may be different from the technically feasible one described herein. The purpose of this exercise was to determine if the project itself was feasible at all. Future concentration on the requirements of the system will be made and an optimal architecture will be adopted.

6 Visibility

The team will take efforts to maximize the visibility of the system and the development process. This will ensure that the project is being developed in line with client specifications. Any deviations from those specifications can also be caught early and corrected through client feedback. Various visibility methods the team intends to use are described below:

Communication

In person meetings and emails would be the primary form of open communication to keep the clients updated with the progress of the project. Regular meetings will be held with the client to discuss progress and for the purposes of two way feedback. The team will also meet as a whole at least once a week in school to assure all members are caught up and understand their roles and jobs. We also use other tools like Airtable or Gantt Chart to manage the progress of each member in the team everytime to make sure no one is behind schedule.

Project progress tracking method

At the end of each milestone, the team will send a presentation and demo product to the client via the Assignment on Microsoft Team. The client can also track the progress via Github or some other online tools. Anything new about the source code would be updated clearly on Github that the client was added in.

Furthermore, you can track our progress by emailing us via Outlook. We are online 24h per day to check it.

7 Risk Analysis

As with any project, this undertaking is not entirely risk-free. Below information is the detail about some risks that we have identified:

Time Risk

As course requirements specify that the project must be completed within one academic semester, any extensions are not possible. This introduces the risk that the system may not be completed with the full functionality the client wants within the given timeframe of a semester. In this case, there is also a second risk of delayed implementation of the system if the client chooses to wait until the system is fully functional.

Technique Risk

No one in our team has experience about backend development so we need to learn and implement simultaneously. This follows the risks that the backend source code may be not professional and contains bugs. However, we will try our best to gain experience as soon as possible and reduce these risks.

No one in our team has experience with website security. We need to learn and practice at the same time to ensure the information of customers is absolutely secure. At least, we can avoid some common attacks like XSS, SQL Injection, DDos but may be not all.

Resource Risk

Resource risks involve technologies the team has available for their use. Due to costs and other external constraints, the team may not be able to obtain the needed or best resources to complete parts of the system. For example, one identified resource is that team members will need laptops running a Linux or Mac operating system to work with servers. However, no one in our team has a laptop like that so we need to try to deal with servers on Windows which is not the ideal software for developing servers.

In addition, there are inherent risks in the resources, such as the software and hardware the team decides to use. Currently, to keep costs at a minimum, the team is considering open-source software, which is available without charge. Risks that stem from resources include hardware failures, system crashes, bugs in the code, etc., which may cause accidental data loss. Since part of the system is web-based, slight variations in display of the user interface may also occur due to different internet browsers.

Functionality Risks

Functionality risks have to do with how the system works. Issues that fall under this category include developing a user interface that is not user-friendly or not well-liked by the client, or producing functions that have limited sustainability. The biggest risk comes from developing a system that does not do what the client wants it to do. Luckily, functionality risks are the easiest to reduce since functionality constraints are more flexible than time or resource constraints. However, minimizing functionality risk is usually accomplished by omitting specific parts and/or functions of the system, as decreasing functionality naturally decreases its associated risks. The team would like to avoid doing this as much as possible. The clients must be aware that it is possible that this must be done in order to deliver the system by the due date at the end of the semester.

Risk Management/Minimization

Having outlined the basic risks associated with this project above, the team is prepared to take precautionary actions to minimize these risks. The principal plan is to develop and practice good management strategies. The team intends to divide the project into a series of iterative phases that have concrete milestones as discussed in previous sections. These milestones will provide project visibility and allow the client to see the team's progress at each stage. Frequent communication and feedback from the client are also essential for client satisfaction with the user interface and functionality. The team will also constantly review their progress and modify goals if necessary to deliver a satisfactory system on time to the client.

8 Business consideration

There are several business considerations that must be taken into account when determining the feasibility of the Tinkerbell Garden project, including but not limited to: disclosure of trade secrets and sensitive information, copyright and trademark issues, and considerations with regards to patents.

Trade Secrets and Sensitive Information

As far as the team could gather from discussions with representatives from the Tinkerbell Garden, the data stored on the database is the information of customers visiting the park which can be considered as the trade secrets. Although it doesn't contain sensitive information, only the phone number, name and age of the customers, it can still be used to develop strategies to improve and attract more customers to the amusement park. Therefore this information must be secret and we have designed the password protected pages to prevent malicious users from accessing and corrupting this valuable data.

Copyrights and Trademark

Since this project is being completed for the Tinkerbell Garden, the team intends to give Tinkerbell Garden a limited license to use and modify the system. The team will not be responsible for any modifications after the software system is delivered, but will help with any questions or concerns of the client as time and circumstances permit. The team reserves the right to be able to demo the software system to prospective employers and showcase the software system as a work created by each team member. Every information about trademarks in the software is provided by the client and we are not responsible for any information which is illegal to appear in the software.

Patents

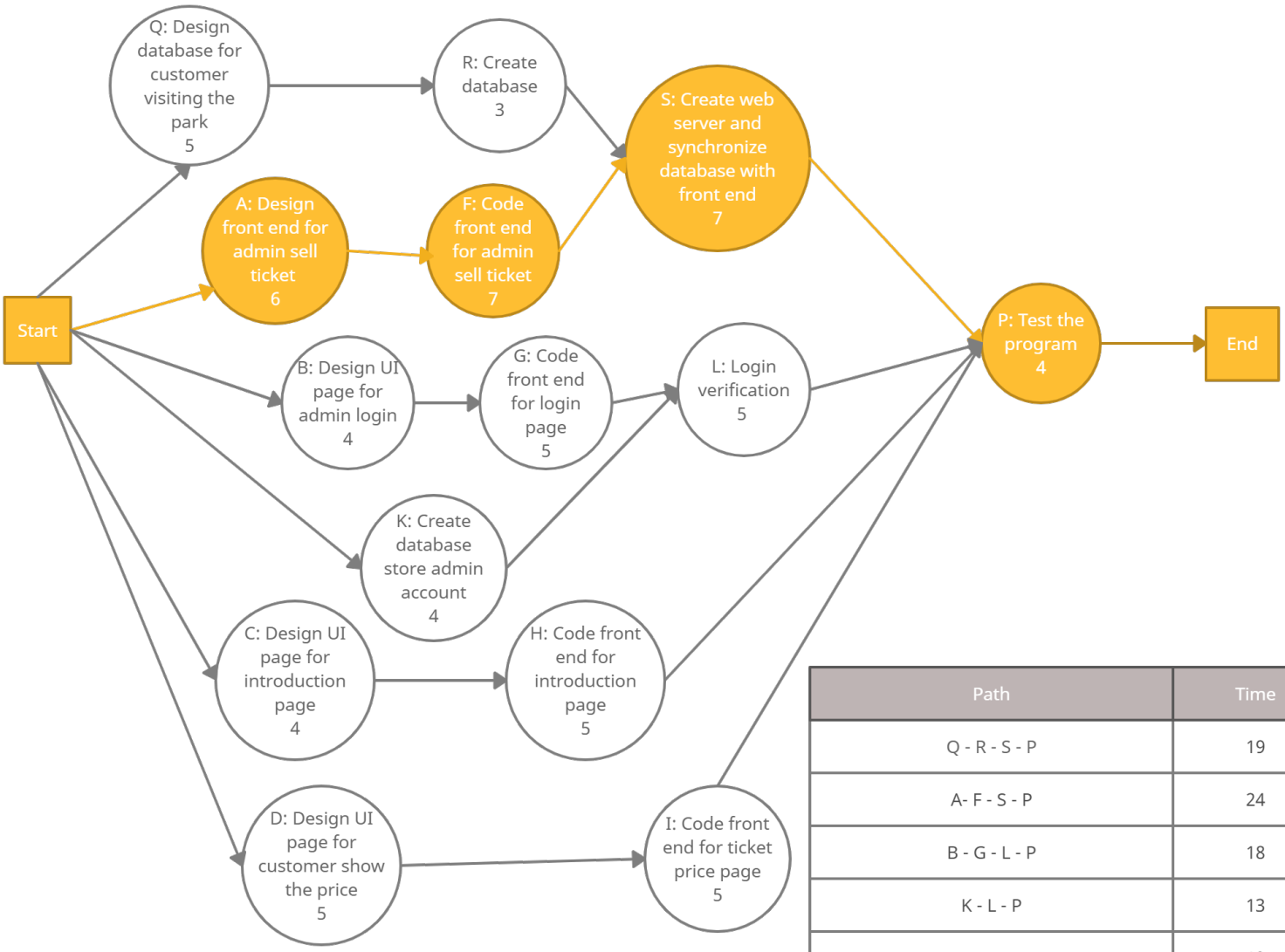
No part of the system is foreseen to be eligible for any patent applications. However, if upon a later date, a part of the system is found to be patentable, the team reserves the rights to the uncontested patent and any derivative works based therein, while the client will automatically gain non-exclusive rights to use the system, and will have full rights to the use and modification of the system regardless of any patent rights held by the team.

9 Conclusion

From the results of the feasibility study, the team finds that the Tinkerbell Garden project is feasible in terms of technicality, skill of team members, and time. Given the time constraint of one semester, the team believes the scope of the project is manageable and that the client's requirements can be satisfactorily fulfilled upon system completion. The team members also possess the adequate skills to implement the system and are familiar with hardware and software that may be used in this project. The conclusion of the feasibility report is to go ahead with this software development project.

10 Schedule for the first sprint

This following preliminary Critical Path Method is used for an approximation of our development process and task schedule for the first milestone of the projects:



Path	Time
Q - R - S - P	19
A- F - S - P	24
B - G - L - P	18
K - L - P	13
C - H - P	13
D - I - P	14

Figure 2. Critical Path Method Graph for the first sprint

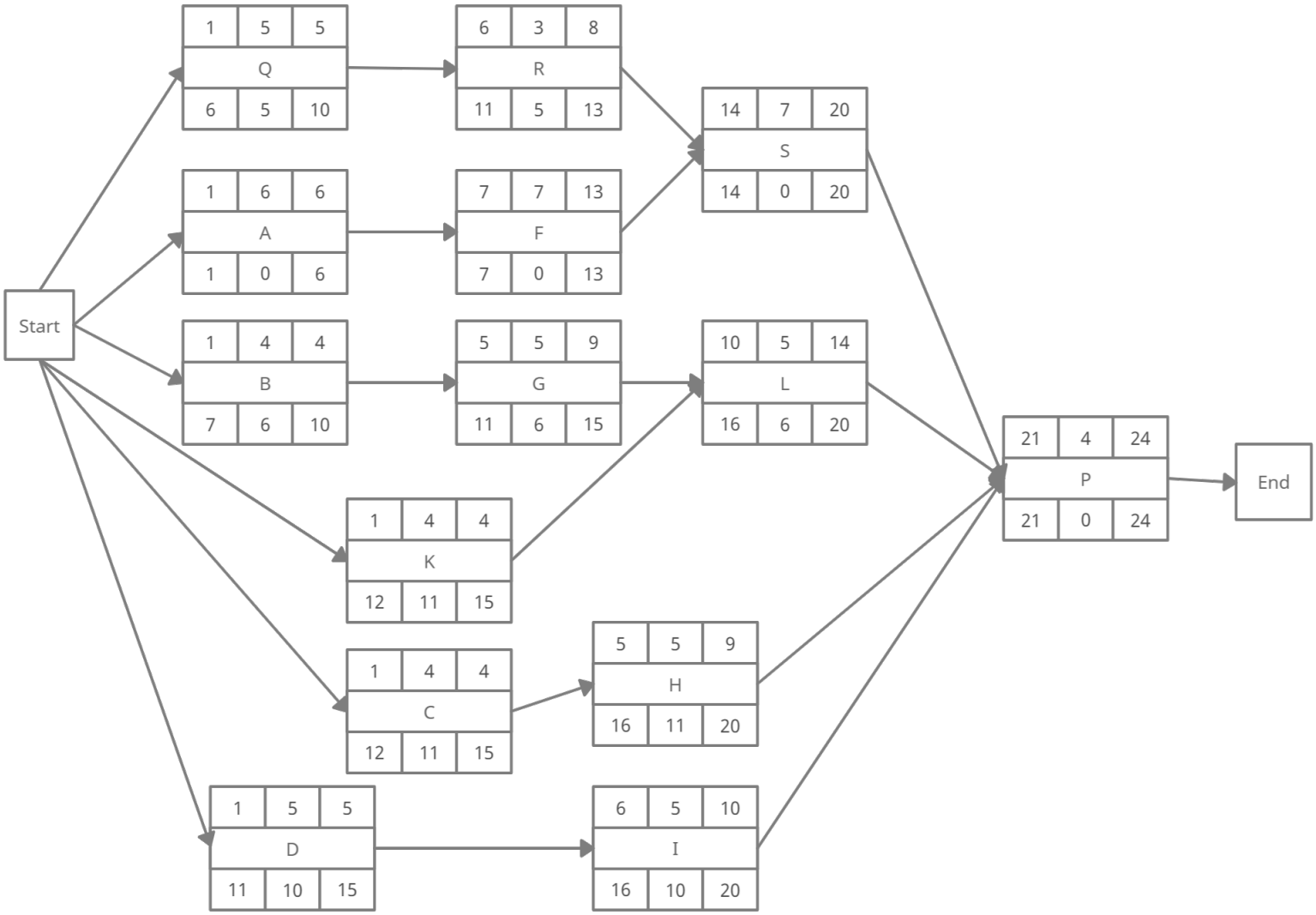


Figure 3. Time estimation for the first sprint