Delphi: SE 491 Final Report

April 20, 2025

Team 15 - Fleure:
Jade Forrest, j6forres
Edward Lee, e237lee
Sam Nolan, s5nolan
Tony Jiang, z273jian

1 Table of Contents

2 Abstract

2.1 Purpose

Online gambling has become a widespread issue due to its addictiveness and high availability. The consequences can be severe, ranging from financial ruin and broken families to intense psychological stress. Even professional athletes have fallen victim to it, with harsh penalties following (Purdum 2021 & Wyshynski). Moreover, various applications designed for investing or gaming often encourage addictive spending habits through mechanisms like instant trading and loot boxes (Cueto 2025, Liberto 2025).

Delphi is a simulated gambling application that allows users to place bets on real-world events, including sports games, political outcomes, and economic indicators. Unlike traditional gambling platforms, Delphi provides a risk-free environment where users can engage in prediction-based activities without the fear of financial loss. This makes it an ideal tool for individuals looking to educate themselves about probability, market dynamics, and strategic decision-making.

Additionally, Delphi may serve as a helpful resource for those struggling with gambling addiction. By offering a controlled, non-monetary alternative, it can act as a stepping stone toward reducing reliance on real-money betting—similar to how nicotine patches assist smokers in gradually overcoming their addiction. Through responsible engagement with Delphi, users can develop healthier habits while still enjoying the thrill of prediction-based gaming.

2.2 Terminology

The following terms appear throughout the report:
Delphi - The name of the application.
User - The person who uses Delphi.
Event - A real life scenario with multiple outcomes that the users will be trying to predict.
Option - The different outcomes for an event that the user can bet on (or against).
Share - A unit representing a user's stake in a particular option within an event. There are Yes shares, which indicate the belief that the associated option will occur, and No shares, which indicate the belief that the associated option will not occur.
Credit - The free currency used in the application to purchase Shares.

2.3 General Overview

Upon creation of an account, users start with 5000 credits (the in-game currency). They can spend credits buying shares, which represent outcomes to events. The price of a share rises and falls with demand, based on the proportion of total shares bought for that option compared to the

other options in the event. After buying, users can sell early or hold on until the event ends, upon which they will be rewarded with 100 credits for winning and nothing for losing.

3 Requirements Specification

3.1 Functional Requirements

Delphi has a number of functional requirements that the app must support in order to operate. Administrators must have the ability to create betting categories, as well as define events and their options, including uploading the corresponding images for those options. Administrators must also be able to release their created events, as well as fetch their past events that have expired. Once an administrator identifies an expired event, they must be able to select a winning option and close the event.

There are other requirements for users of Delphi. Some requirements are for all users, regardless of whether they are authenticated or not. They should be able to read information about gambling, the risks involved, and how to gamble safely. Similarly, all users should be able to see all open events and their options, as well as the leaderboard.

If not logged in, users should be able to create an account using a unique email and username. A user must also be able to log into their account using their email and password.

Once logged in, a user should have the option to log out. They should also be able to access their account information, including their bets and credits. Furthermore, authenticated users with sufficient funds must be able to purchase shares for any open events. They must also be able to sell any shares they have purchased. Upon purchase or sale of shares, the shares-owned list and credits of the user must be updated.

The system should recalculate the option prices in a timely manner (at least every 5 minutes). The system should not be used to automatically close events as it cannot automatically identify the winning option.

3.2 Non-Functional Requirements

Delphi has a range of non-functional requirements that are crucial to the overall success and effectiveness of the product. Security stands as one of the most important aspects of the application, as it is essential to protect the user's personal information. All user data, especially login credentials, must be securely encrypted to ensure that unauthorized access to user accounts is prevented. This level of protection is necessary not only to comply with privacy standards but also to build trust with users who rely on the app to safeguard their sensitive information.

In addition to security, performance is a key factor in the success of Delphi. The app must deliver fast and responsive interactions to ensure that users can make informed decisions quickly,

especially when it comes to assessing their current standings and placing bets. This means that the system must be optimized for low latency, with quick response times under all conditions. A slow or laggy app results in a poor user experiences and potentially lost opportunities, making this requirement critical.

Usability is another cornerstone of the Delphi app. It is important that users can intuitively navigate the app and access all its features without confusion. The design should be user-friendly, with simple yet effective navigation, clear instructions, and easy-to-understand actions. A seamless and intuitive experience will reduce user frustration and increase engagement, ensuring that people can easily access the app's features without a steep learning curve.

The system must also be able to handle varying levels of user activity with predictable performance. This includes managing high traffic volumes or sudden changes in events without causing disruptions. In scenarios where there is a spike in the number of active users, the app should maintain near-optimal performance. The system should be robust enough to manage a large number of concurrent users and ensure data consistency across users. This includes providing real-time updates on important metrics such as share prices, leaderboard rankings, and current bets, ensuring all users have access to the same accurate and up-to-date information.

Furthermore, Delphi's system must be resilient enough to operate without errors, failures, or unexpected behaviors. It is critical that the app performs all of its functions reliably, with high availability and minimal downtime. Users should never encounter issues that prevent them from accessing key features or completing tasks, as this could seriously undermine the app's reliability and user satisfaction.

Regarding accessibility, Delphi should be available on mobile devices, both iOS and Android. This is because it must imitate gambling, which is often a spontaneous and on-the-go activity for many users. The mobile platform allows users to place bets, track their standing, and interact with the app from virtually anywhere, offering the flexibility and convenience they expect from modern betting applications.

3.3 System Requirements

The images associated with options for an event must be stored in a manner that allows for efficient retrieval. Since these images are not associated with sensitive personal information, there is no need for strict security measures to protect them from unauthorized reads. However, it is critical that these images are safeguarded from modifications.

On the other hand, data regarding events and users must be stored securely. Personal information, login credentials, and any other data associated with users must be protected from

unauthorized access and breaches. Ideally, the storage and retrieval of this information should be free, easy to integrate, and user-friendly.

3.4 User Stories

3.4.1 User Stories for Administrators

Create Betting Categories: As an administrator, I want to create betting categories so that I can organize different types of events for users to participate in.

Define Events and Options: As an administrator, I want to define events and their options, and link corresponding images so that users can see and bet on those options once released.

Release Created Events: As an administrator, I want to release created events so that users can start placing bets on them.

Fetch Expired Events: As an administrator, I want to be able to fetch past events that have expired so that I can review and manage them.

Select Winning Option: As an administrator, I want to select a winning option for expired events so that the event can be closed.

Close Event: As an administrator, I want to close an event so that bettors can receive their winnings.

3.4.2 User Stories for Non-Authenticated Users

Create an Account: As a non-authenticated user, I want to create an account using a unique email and username so that I can participate in betting and track my activities.

Log Into Account: As a non-authenticated user, I want to log into my account using my email and password so that I can participate in betting and track my activities.

3.4.3 User Stories for Authenticated Users

Log Out: As an authenticated user, I want to log out of my account so that I can secure my personal information and prevent unauthorized access.

Access Account Information: As an authenticated user, I want to access my account information, including my bets and credits so that I can keep track of my activities and funds.

Purchase Shares: As an authenticated user with sufficient funds, I want to purchase shares for any open event so that I can participate in betting and potentially earn rewards.

Sell Shares: As an authenticated user, I want to sell any shares I have purchased so that I can increase my amount of credits.

3.4.4 User Stories for Both Non-Authenticated and Authenticated Users

Read Gambling Information: As a user, I want to read information about gambling, its downsides, and how to gamble safely so that I can make informed decisions.

View Open Events and Leaderboard: As a user, I want to see all open events and their options, as well as the leaderboard so that I can stay updated on ongoing events.

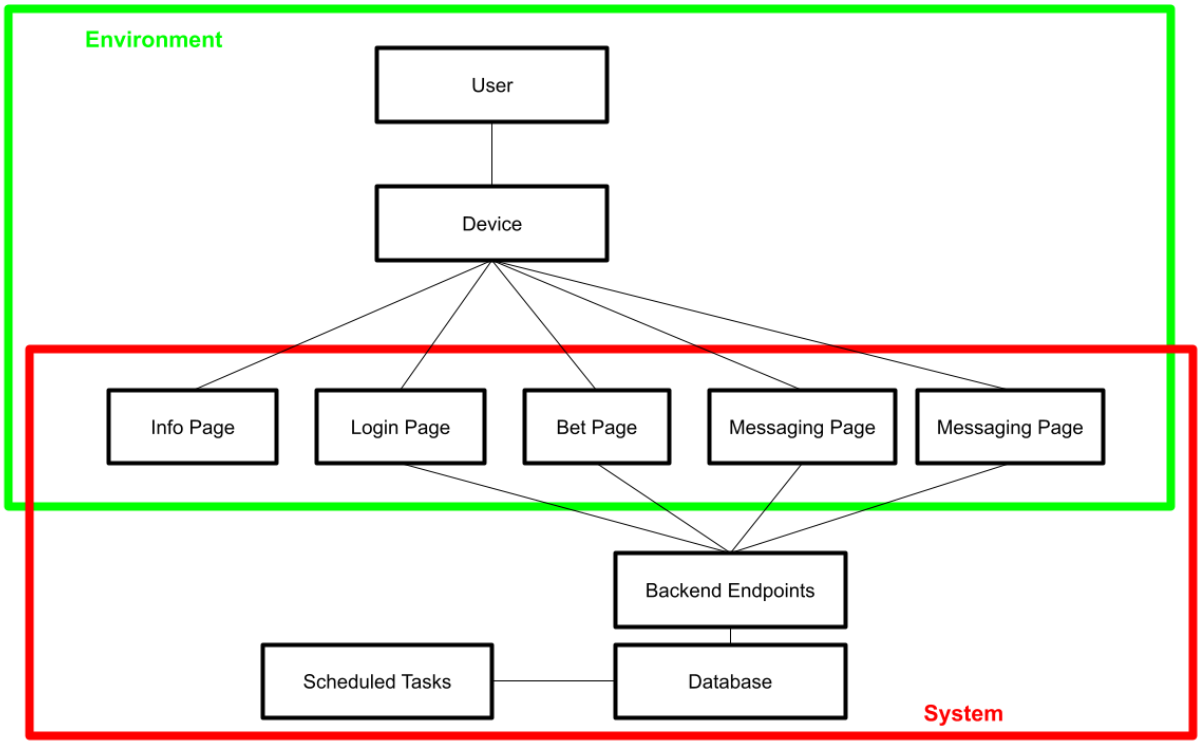
3.4.5 User Stories for System

Read Gambling Information: As the system, I want to refresh the share prices for all events at regular intervals to accurately reflect their market value, ensuring that users always have access to the most current price.

4 Design Documents

4.1 Domain Model with Superimposed World Diagram

This domain model is derived from the requirements. The plan is to have 5 main pages for the user to interact with through their device, calling endpoints that query the database. There will be scheduled tasks to update the share prices regularly.



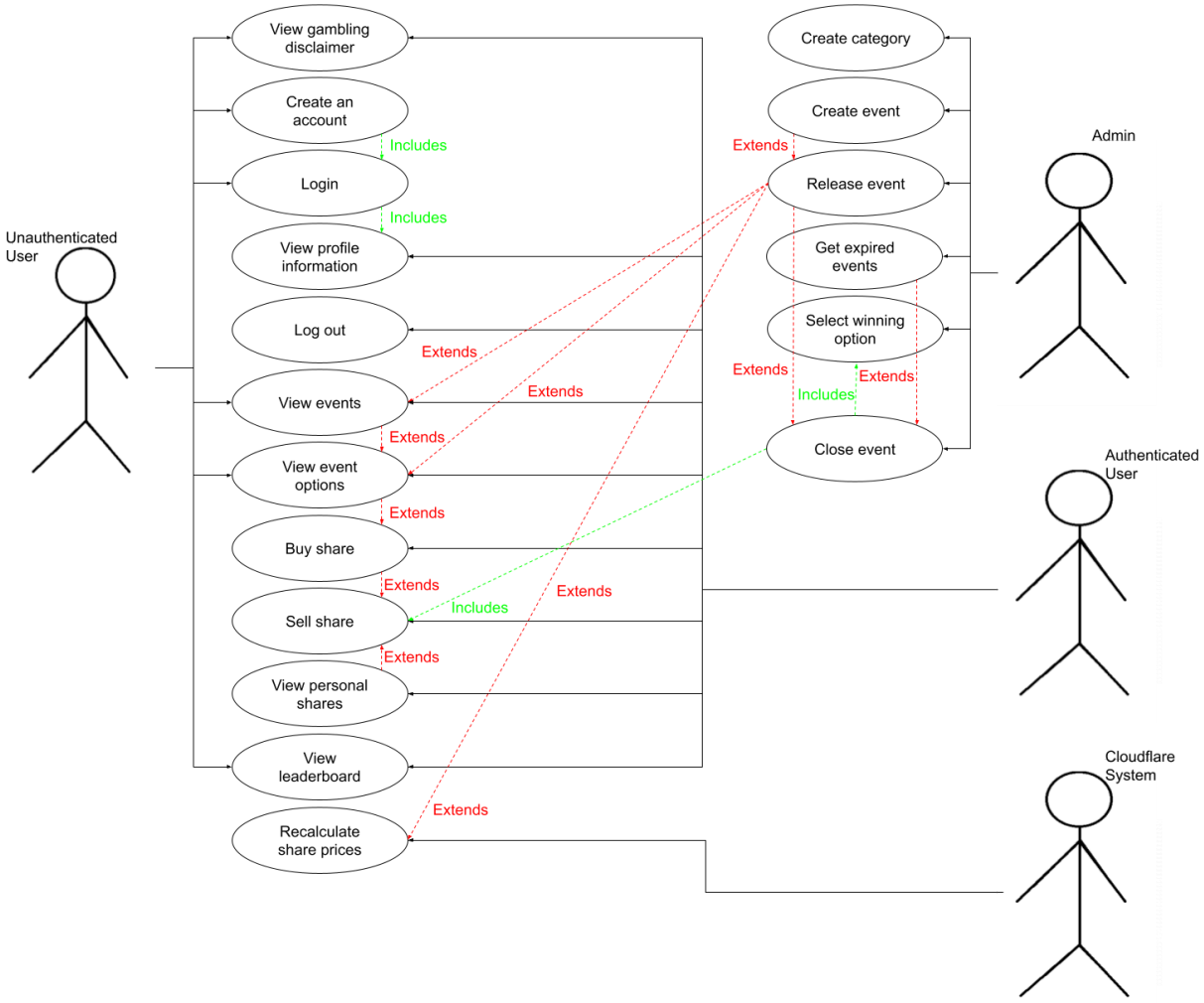
4.2 Technologies and Cloud Plans

In line with our system requirements outlined in section 3.3, we opted to upload the option images to Imgur. For hosting our SQL database and backend, we selected Cloudflare because of its generous free tier, easy-to-use interface, and its policy of blocking extra requests rather than charging for exceeding the free limit.

For the frontend, we decided to use Flutter, due to its flexibility to work for both iOS and Android, as well as for a website.

4.3 Use Case Diagram

The use case diagram is derived from the user stories in section 3.4.



4.4 API Documentation

From the use case diagram, we can derive specifics for the endpoints. Details like parameters and return value can be documented.

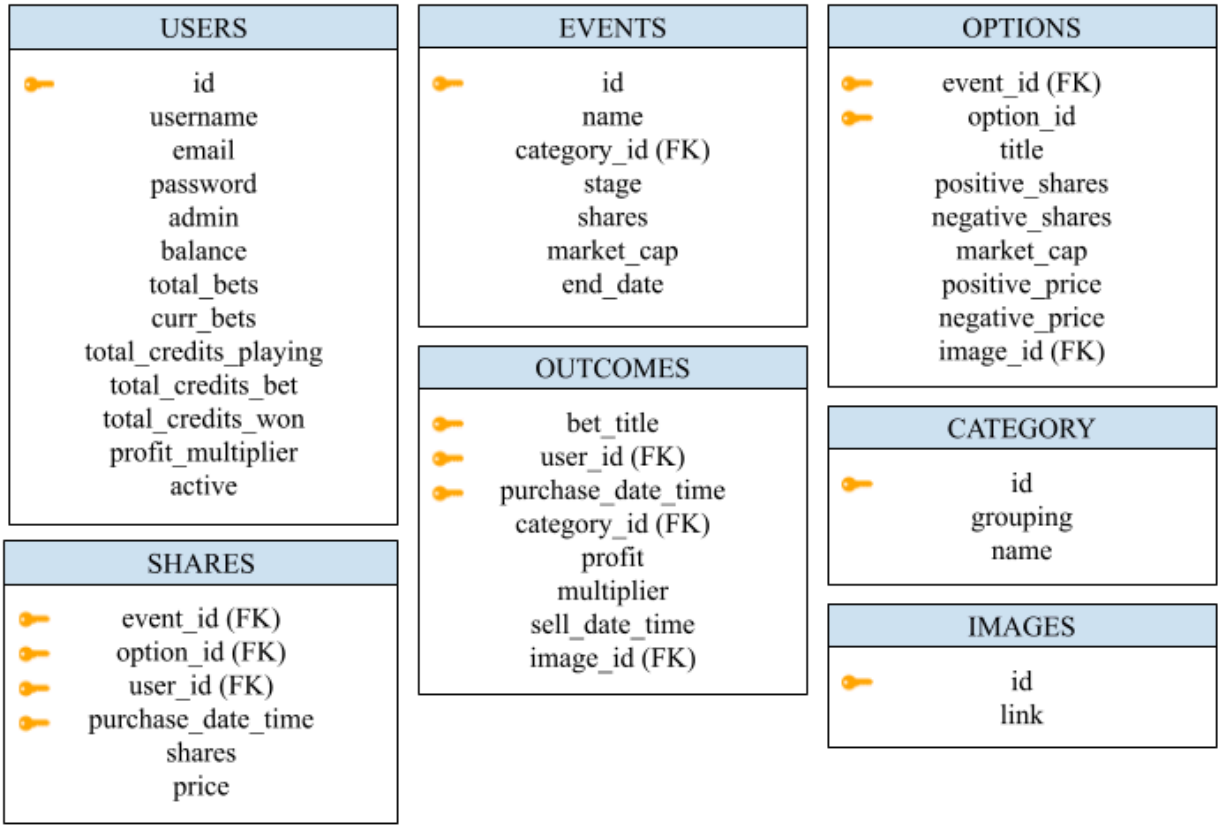
Worker	**Endpoint**	**Method**	**Input**	**Output**
user	create_account	POST	username: string, email: string, password: string, admin: bool? balance: int?, total_bets: int?, total_credits: int?, total_credits_won: int?, profit_multiplier: int?	success: bool
user	login	POST	username: string, password: string	success: bool, id: int
user	get_profile_details	GET	id: int	success: bool, username: string, admin: int, balance: int, bets_count: int, total_bets: int, curr_bets: int, total_credits_playing: int, total_credits_won: int, profit_multiplier: int
admin	create_category	POST	name: string, grouping: int	success: bool
admin	create_event	POST	event_name: string, cateogory_name: string, end_date: int options: [{name: string, image_link: string}]	success: bool
admin	release_event	POST	name: string	success: bool
admin	get_expired_events	GET		success: bool, events: [{name: string, end_date: int}]
admin	end_event	POST	name: string, winning_option: string	success: bool

buyer	buy_shares	POST	user_id: int, event_name: string, option_name: string, share_count: int	success: bool
seller	sell_shares	POST	user_id: int, event_name: string, option_name: string, purchase_date_time: int, share_count: int	success: bool, revenue: int, profit: int, multiplier: int
categories	list_cateogires	GET		success: bool, events: [{grouping: int, name: string}]
list-events	list_events	GET	categories: [string]?, order_by: string?, order_direction: string?, page: int?	success: bool, events: [{name: string, shares: int, market_cap: int, end_date: int, top_option: string, top_price: int, top_image: string}]
list-events	list_user_events	GET	user_id: int, categories: [string]?, order_by: string?, order_direction: string?, page: int?, bought_only: bool?	success: bool, events: [{name: string, shares: int, market_cap: int, end_date: int, top_option: string, top_price: int, top_image: string, user_bought: bool}]
list-options	list_options	GET	event_name: string	success: bool, options: [{name, string, positive_shares: int, negative_shares: int, market_cap: int positive_price: int, negative_price: int, image_link: string}]
list-options	list_user_options	GET	user_id: int, event_name: string	success: bool, options: [{name, string, positive_shares: int, negative_shares: int,

				market_cap: int positive_price: int, negative_price: int, image_link: string, user_bought: bool}]
list-shares	list_event_shares	GET	user_id: int, event_name: string, order_by: string?, order_direction: string?	success: bool, shares: [{event_name: string, option_name: string, bought_date_time: int, event_end_date: int, shares: int, price: int, current_price: int, Image_link, string}]
list-shares	list_shares	GET	user_id: int, categories: [string]?, order_by: string?, order_direction: string?, page: int?	success: bool, shares: [{event_name: string, option_name: string, bought_date_time: int, event_end_date: int, shares: int, price: int, current_price: int, Image_link, string}]
list-outcomes	list_outcomes	GET	user_id: int, categories: [string]?, order_by: string?, order_direction: string?, page: int?	success: bool, outcomes: [{bet_title: string, bought_date_time: int, profit: int, multiplier: int, sell_date_time: int, image_link: string}]
leaderboard	list_leaders	GET	order_by: string?, order_direction: string?	success: bool, users: [{username: string, balance: int, total_credits_won: int}]
recalculate	recalculate_prices	POST		success: bool

4.5 Database Schema

The database schema was derived to fulfill our functional requirements in 3.1.



4.6 Frontend Figma

A few of the important pages were designed and prototyped in Figma: https://www.figma.com/proto/UwTbyY4l5qi9bIWKZ6Ju4L/App-Design?node-id=7-1133&t=KABlmxR3dn9b83Pu-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=7%3A1133

5 User Manual and Access Information

5.1 User Manual

5.1.1 Info Page

5.1.1.1 Read Gambling Disclaimer

Navigate to the first tab.

5.1.2 Account Management

5.1.2.1 Login

Must not be logged in. Navigate to the second tab, enter the email and password in the appropriate fields, then click the Login button. If an account has not been made yet, please reference section 5.1.2.3.

5.1.2.2 Logout

Must be logged in. Navigate to the second tab, and click the Logout button.

5.1.2.3 Account Creation

Must not be logged in. Navigate to the second tab, and click the Sign Up text. Then in the account creation pane, enter the desired username, email, and password, and click the Sign Up button.

5.1.3 Bets and Shares

5.1.3.1 Checking Available Shares to Buy

Navigate to the third tab, a list of currently available events will be shown. Each row for their respective event will contain the title, leading choice from other users, total amount of credits in the pool, total shares bought in the specific event, share price for the YES share of the leading choice, and the expiration date of the bet.

Click on any event to see all the available options for a given bet, and the price for YES and NO shares for each option.

5.1.3.2 Buying a Share

Must be logged in. Navigate to the share you want to buy using section 5.1.3.1. Click on the YES/NO button for the share you want to buy. A pop-up will be shown and enter the desired amount to buy. To confirm, click the Buy button.

5.1.3.3 Viewing the Owned Shares for an Event

Must be logged in with shares for the event owned. Navigate to the third tab and click on the event that correlates to the share. Then, click the My Shares button to see every share owned for this event.

5.1.3.4 Selling a Share in the Bet Page

Perform the steps in 5.1.3.3 and scroll to find the share in question. Click on the red or green button for that share (red if a YES share, green if a NO share) and enter the number of shares to be sold. Then confirm by clicking the blue SELL button.

5.1.3.5 Viewing All Owned Shares

Must be logged in with shares owned. Navigate to the fourth tab. For each share, the current price, date of purchase, expiration date for the bet, and difference between bought/current price are shown.

5.1.3.6 Selling a Share in Shares Page

Perform the steps in 5.1.3.5 and scroll to find the share in question. Click on the share and a pop up will appear. Enter the number of shares to be sold, and confirm by clicking the Sell button. When already logged in, navigate to the “Shares” tab, where a list of bought shares will be shown.

5.1.4 Leaderboard

5.1.4.1 View Leaders with the Most Credits

Navigate to the fifth tab. On the second row, there are 2 boxes with the words Balance and Winnings. If Balance is not in a blue box, click the box.

5.1.4.1 View Leaders with the Most Winnings

Navigate to the fifth tab. On the second row, there are 2 boxes with the words Balance and Winnings. If Winnings is not in a blue box, click the box.

5.2 Access Instructions

5.2.1 Web

While not originally intended for the browser experience, a web version of Delphi is deployed on Cloudflare. It can be accessed via this link: https://delphi-oracle.pages.dev/

5.2.2 Android

For android users, the apk file for Delphi is publicly available and can be downloaded here: https://drive.google.com/file/d/1hbg16udk4sdQGv8Efv3FuJaX-VSITUH5/view?usp=sharing

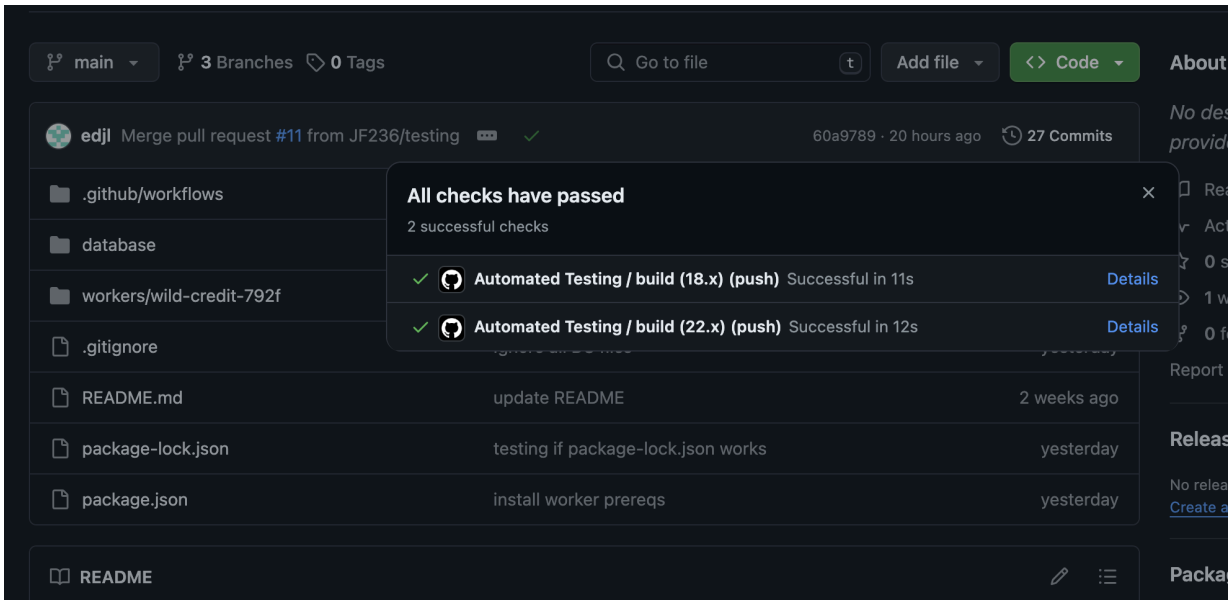
5.2.2 iOS

Unfortunately, the iOS version of Delphi was rejected by Apple from their App Store because it contains simulated gambling. Their terms and conditions only allow companies to deploy such applications.

6 Verification and Validation

6.1 Testing Strategies

Unit tests were implemented to verify the backend and ensure the proper functionality of the endpoints. To prevent regression, we integrated them into our CI/CD pipeline, which automatically verifies that new commits do not introduce regressions.



6.2 User Feedback

User feedback played a crucial role in determining which new events to add. For instance, many events related to economic indicators were added based on popular demand.

Feedback also led to the development of new features. Users requested a notifying indicator for events they placed on, which was implemented. Later, users asked for the option to see their bets for a specific event after clicking on an active event, leading to the addition of a My Shares button and panel for each event. As feedback continued, users expressed the need to sell shares from the My Shares panel, which was implemented.

User feedback for Delphi can be submitted here:
https://sites.google.com/view/oracledelphi/feedback

There may be some users that have concerns about Delphi being a gateway into gambling. This is a valid concern. For example, vaping as an alternative to smoking has been shown to increase the rate of cigarette usage (Chatterjee 2018). This is at odds with studies that clearly show that

vaping is far healthier than smoking (Daiber 2020). This raised a very important question: is Delphi's mission ethical?

As with most ethics questions, the answer is "it depends". Delphi is not the only "simulated gambling" app on the market. Other apps already make online gambling easily accessible. Delphi's uniqueness comes from its complete lack of a profit model. Without an intention to make money off gamblers, Delphi poses less of a threat than other similar products that allow you to spend real money on continued play. In conjunction with the free-to-play environment, Delphi offers an education experience to users. Furthermore, Delphi has a limit on the number of active bets a user can have at one time, ensuring addictions are not easily developed.

In the current mobile app culture, users are already exposed to more subtle and dangerous forms of gambling, with these features integrated into popular mobile games like Clash Royale, FIFA, Pokémon Go, and others. These games often incorporate mechanics that mimic gambling, such as loot boxes or randomized rewards, which have become increasingly normalized.

6.3 Bug Fixes and Iterations

Continuous improvements to Delphi have been driven by both user feedback and ongoing testing. Identified bugs were fixed based on their impact on user satisfaction and the functionality of the app.

In addition to bug fixes, new iterations of Delphi have focused on refining features based on usage patterns and requests from the community. Enhancements have been made to the app's performance, reducing load times and ensuring smoother transitions between event views. The interface has also been updated to improve usability, making it more intuitive for users to navigate the various features.

Each iteration of the app after fixing bugs and implementing new features is tested with the unit tests mentioned in 6.1, and manually tested by the development team. Then it is tested by a dozen volunteers across different devices and operating systems before being released. The development team remains committed to making continuous improvements, with regular updates that address both functional and user-experience challenges.

7 References

Chatterjee K., Alzghoul, B., Innabi, A. & Meena, N. *Is vaping a gateway to smoking: a review of the longitudinal studies*. International Journal of Adolescent Medicine and Health, 30(3), 20160033. 2018. https://doi.org/10.1515/ijamh-2016-0033. Accessed March 26, 2025.

Cueto, J.. *Loot box purchases linked to gambling and video game addiction, study finds*. Theeducatoronline.com; The Educator K/12. March 20, 2025. https://www.theeducatoronline.com/k12/news/loot-box-purchases-linked-to-gambling-and-video-game-addiction-study-finds/286799. Accessed March 26, 2025.

Liberto, D.. *Is Warren Buffett Right That the Stock Market Is Like a Casino? What Investors Need to Know*. Investopedia. 2025. https://www.investopedia.com/warren-buffett-stock-market-like-a-casino-8781412. Accessed March 26, 2025.

Purdum, D. & Wyshynski, G.. *Evander Kane betting allegations: What we know and what we don't*. ESPN. August 2, 2021. https://www.espn.com/chalk/story/_/id/31944608/evander-kane-betting-allegations-know. Accessed March 26, 2025.

Daiber, A., Hahad, O., Kuntic, M., Münzel, T. & Steven, S.. *Is vaping better than smoking cigarettes?*. European Heart Journal, Volume 41, Issue 28, 21 July 2020, Pages 2612–2614, https://doi.org/10.1093/eurheartj/ehaa267 Accessed March 26, 2025.