**Application Launch Plan**

After a long development process full of sweat, tears, and blood the time has come to launch the app in the store. The success of the app depends on the features that will address the issues or gaps from other applications, demand and timing. A good time to launch the app would be around New Year as everyone is giving themselves resolutions to fulfill in the upcoming year. This app is bringing features that are simple but work well and allow users to track their weight without all the annoying clutter of information and information overload.   
  
What will I include in the description for the app would be something like: “Are you tired of all the applications with too complicated features that you don’t need and cause you to suffer from information overload? Then look no further, Weight Tracking App is here to help you reach your goals and aim for the stars! You know what they say, Sky is the limit. With features such as adding, updating, or removing your daily weight and seeing how you are approaching your goal, this app will help you follow your journey and stay on the correct path. With its simplistic design you don’t have to worry about getting lost in the application’s features, while still having the most important ones at the palm of your hand.” I believe this would be attracting to users who are searching for an app that is simple and will fulfill the need to track user’s weight. Me, myself, I sometimes get lost in those complicated applications and get discouraged with all the information that I am seeing.

As per the class’s requirements, I developed the app using version 33 of Android also called “Tiramisu”. After reviewing a website called apilevels.com, I found out that around 30.33% of users are on this version of android. Since this is one of the newer versions of android (There is only version 34 above it), this version is not as commonly used as for example version 27 and lower which all have above 90% cumulative usage. These newer versions, however, have better security and features which makes it more lucrative for future app updates to improve user experience and ratings of the app. This comes with a downside as not as many people will have the option to download my app with lower version levels. Maybe an option would be to go one or two versions lower to get a bigger user platform (Belinski).

For reference see the screenshot below.

A screenshot of a computer

Description automatically generated

The only permission that the app will require is allowing to receive text messages, but only if the user allows and adds a phone number at the registration screen. This will allow the app to send a message once the user reaches their goal weight. Otherwise, no other permissions are needed for the app to run smoothly. If user selects deny texts, they will give the app zero permissions.

I will not monetize my app with a one-time purchase. This does not seem a feasible and scalable option as you require a lot of downloads and purchases to happen to get good money out of the app. The way better option as I talked about in my discussion is the gaming applications way of monetization. I have some free spaces on my layouts where I could add ads to generate revenue. These ads usually pay less than monetizing the app but with a stable user fanbase these small fees could go up very fast. There is also the option to show un-skippable ads occasionally, to increase revenue but these get annoying and could cause users to uninstall the app. One more option to consider is a monthly subscription fee that would offer no adds and special features other users won’t get. This option would take time as there will need to be some updates to app to offer special features for subscription users which the app does not have yet.

**References**

Belinski, E. (n.d.). Android API Levels. API Levels | Android versions, SDK/API levels, version codes, codenames, and cumulative usage. <https://apilevels.com/>