UPDATING THE DAYFORCE HCM APP

SUBMITTED BY: JEEVAN BAINS
SUBMITTED TO: SAM LEE
SUBMITTED ON: APRIL 1, 2018

Table of Contents

List of Figures

List of Tables

Summary

This proposal focuses on adding key features that are currently unavailable within the Dayforce HCM mobile application. Dayforce is a scheduling platform used by many retail stores, and it is used for everything having to do with shifts, payroll, and more.

The current version of the mobile application allows employees to view their schedule, but only for the current and previous weeks. It is also missing many features already available in other scheduling platforms including a shift trading option, a useful display of future availability, and upcoming event information.

The updated version of the application will have a broader view of upcoming shifts, add a shift trading option, a more useful availability tab, and added special event information for upcoming shifts. The application will be updated on both android and ios.

To carry out this project we will need a team of six members, and the project will be broken down not four phases. The entire project will take up to fourteen weeks to complete, including development and quality assurance. The total cost for the update will be $33,480 for updating both the android and ios application.

Phase one will focus on rebuilding the database to display more available shifts. Phase two will consist of adding the missing key feature of shift trading. Phase three will be focused on refining the availably tab currently available and adding useful functionality to it. The fourth and final phase will add event information to applicable shifts and have a final quality assurance test before releasing the update.

The added functionality and ease of use of this update will save management valuable time, have employees be better prepared for their upcoming shifts, and allow employees to have an much more pleasant experience when trying to get a shift covered by a fellow employee.

We recommend starting as soon as possible, allowing users to get familiar with the updated application and allowing them to start enjoying the new useful features.

Introduction

Currently the Dayforce HCM app used for scheduling at Sportchek retail stores is missing a few key features, that are available in other scheduling platforms. These include slow schedule updates, lack of shift trading options, lack of shift information, as well as a somewhat useless availability tab, that can be greatly improved.

For the lack of features stated above, I am proposing on adding many needed features to the app and refactoring the code so elements that are available from different platforms are available when needed.

My proposal would bring many useful features to the already existing app. These features include refactoring the show schedule portion of the app to display shifts already input in the database, adding a shift trade option for employees to directly trade shifts with the manager being able to approve the request from his login, adding special event information if necessary for certain shifts, as well as altering the availability tab to add some much needed functionality.

These changes would enable employees to have a better view of their upcoming schedule, it would save the managers time when updating schedules by not having to manually enter changes when updating the schedule, it would allow employees to be more prepared for there upcoming shifts for special events, as well as make it easier and more convenient for employees to find a replacement for any shifts they may need covered.

Proposal Details

The update of the Dayforce HCM app will be split into five phases, four of these phases consist of a different feature to be added or updated, while the fifth phase consist of a quality assurance test to make sure all features added work well and as planned. In order to accommodate all staff members this update will be done to both the android and ios version of the app.

For these stages to be carried out we will need a team of six key members, including both an android and ios developer, a quality assurance coordinator for each platform, an application administrator, as well as a backend database developer.

Both the android and ios developers will be responsible for carrying out the necessary changes for each platforms application. These developers would create a modified version of the app after adding each new feature to the app that would allow for the quality assurance members to test the functionality.

The quality assurance members will test each new feature of the app as it is added, as well as a final version of the app with all features working at the same time.

The backend database developer will be responsible for adding the necessary changes that allow for employees to directly trade shifts and mangers being able to approve those changes and making these changes final and appear in the new updated schedule.

Our final member needed for carrying out this update would be a application administrator, who will be responsible for installing, configuring, and updating the software that is in the retail locations, so that they will work seamlessly with the new updated applications.

Phase One

Phase one will consist of the database developer and app developers working together to allow employees to view shifts already input into the system. This phase would focus on one aspect of updating the app, and lead into the following phases.

Figure 1 Dayforce App vs Available Schedule
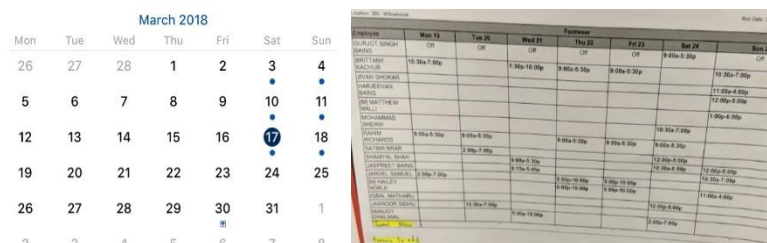
As seen in Figure 1 above the schedule displayed on the application is only able to display shift of the current and previous weeks, where as you can see on the right the following weeks schedule has already been posted and available in the database.

The main changes needed to be made are just for the database developer allowing the app developers to access new week schedules as they come into the database, instead of waiting and creating a delay. After the changes are made, it can be tested by the quality assurance members.

At the end of this phase, the application will have one of the four features implemented and we can now move on to implementing another feature.

Phase Two

Phase two will be focused on adding a shift trade option, so employees can directly offer co-workers a shift and manager can approve the change directly from the app. This would save time because the manager would not have to manually change the schedule whenever a request is made.

This feature would require both the android and ios developer to incorporate an third part API into the app. "This will require extra time to learn the third party APIs and implement them correctly" (Kohan, 2015).

Throughout this phase the developers will have to test that their code works with the existing database, and works seamlessly across both platforms, so that employees of both operating systems may use the feature with one another.

After the feature is developed, the quality assurance members can now test this new aspect as well as the previous version to make sure there are still no bugs. The application administrator will also need to learn and understand the basis of this feature and be able to setup and explain the updated version of this part of the app to managers and employees of the retail stores.

Phase Three

Phase three will consist of building upon the previous 2 phases and using those new features and altering an already available feature to make a seamless shift covering experience. During this phase we will be rebuilding the availability tab as seen in Figure 2 below.
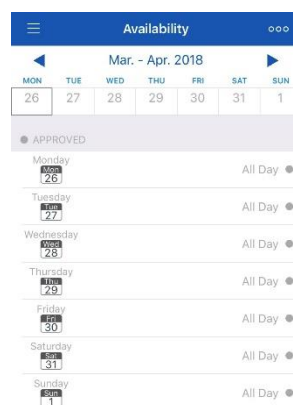

Figure 2 Availability Tab

As shown in Figure 2 above, the current status of the availability tab is unused. Combining first feature of showing multiple weeks schedule, and the second feature of offering shift trading functionality, we will now alter this tab and display coworkers given availability as well as the users, making it even easier to find an employee to cover a shift.

This feature would require the database developer to create a database to store the given availability and allow for it to be changed by the users, as well as the android and ios developers to integrate this new code in their respective platforms.

The quality assurance members will then test this updated feature along with the previous versions to ensure there are no bugs.

Phase Four

Phase four will consist of adding the final feature which will be adding special shift information when necessary. After adding this feature employees will be more prepared and ready for their upcoming shifts, when big promotions are right around the corner.

To make this feature a reality, both the ios and android developer must be able to access the information on when a sale is going to occur. This will require the database developer to create an API for storing the upcoming event information, most likely with NodeJS. "NodeJS is a simple yet powerful programming boon for API making" (Meenakshi, 2017).

After creating the API, the developers will now update the shift information to be displayed, and add in any special event notes if necessary.

After this fourth and final feature has been added, the quality assurance members can now test a final version of the app, with all new features implementing and working. After thorough testing he application administrator will learn about all the new features of the app and present these features to the end users.

Schedule

Updating the Dayforce HCM app will take approximately 14 weeks, where the time of each phase will be different from one another in length as shown in the table below. We will discuss more detail about how long each phase will take and what is needed in those phases after.

Table 1 General Schedule

Phase	Approximate Time for Completion
Phase one	2 ½ weeks
Phase two	3 weeks
Phase three	3 ½ weeks
Phase four	5 weeks

Please note that all of these time estimations are based on data from https://infinum.co/ article on how long it takes to build an app, with slight modifications made to account for differences with the Dayforce app (Kovačević, 2016).

Phase One Schedule

Phase one will be focused on adding the first feature which will require help from both the android and ios developers, the database developer and the quality assurance members. Specific actions will be described in the table below.

Table 2 Phase One Schedule

Phase One Details	# Of weeks to complete
Database developer will rebuild the constraints on the database to allow the developers to access more information	1
Both the ios and android developer will update the app to accommodate for the changes made to the database	1
Quality assurance members will test the new feature	1/2

As shown in the table above this entire phase will take approximately 2 ½ weeks to complete. This is a broad guideline with included buffer space.

Phase Two Schedule

Phase two will be focused on adding the second feature which will require help from both the android and ios developers, the application administrator, as well as the quality assurance members. Specific actions will be described in the table below.

Table 3 Phase Two Schedule

Phase Two Details	# Of weeks to complete
Both the ios and android developer will update the app to accommodate for the changes needed to be made to the apps	2
Quality assurance members will test the new feature	1/2
Application Administrator will learn the ins and out of how this feature works	1/2

As shown in the table above this entire phase will take approximately 3 weeks to complete. The development of this phase will take longer than the other phases, because the developers will need to incorporate a third party API to add this functionality. This is a broad guideline with included buffer space.

Phase Three Schedule

Phase three will be focused on adding the third feature which will require help from both the android and ios developers, the database developer and the quality assurance members. Specific actions will be described in the table below.

Table 4 Phase Three Schedule

Phase Three Details	# Of weeks to complete
Database developer will create a new database to store given availability for all employees	1
Both the ios and android developer will update the app to accommodate for the changes needed to be made to the apps after the new database is created	2
Quality assurance members will test the new feature	1/2

As shown in the table above this entire phase will take approximately 3 ½ weeks to complete. The development of this phase will take longer than the other phases, because the developers will need to incorporate a new database into the system. The database developer will also need more time, because he will be required to create a new database. This is a broad guideline with included buffer space.

Phase Four Schedule

Phase four will be focused on adding the final feature and testing the final version of the application with all new features added which will require help from both the android and ios developers, the database developer, the quality assurance members, and the application administrator. Specific actions will be described in the table below.

Table 5 Phase Four Schedule

Phase Four Details	# Of weeks to complete
Database developer will create a API to store upcoming event information	1
Both the ios and android developer will update the app to accommodate for the changes needed to be made to the apps after the new API is created	2
Quality assurance members will test the new feature and the final build of the app	1
Application Administrator will learn all the new features of the app, and learn the basics of how they work	1

As shown in the table above this entire phase will take approximately 5 weeks to complete. The quality assurance of this phase will take longer than any other, because it is the final version o the app. This is a broad guideline with included buffer space.

Budget

Ball-park Cost Breakdown for **Native App Development: Android & iPhone**						
Project	Small MVP	Small Enterprise	Medium MVP	Medium Enterprise	Large MVP	Large Enterprise
Discovery and Wireframing	$1,500	$2,500	$2,500	$3,500	$5,000	$9,000
App Screen Designs	$2,000	$3,000	$3,500	$5,000	$5,000	$9,000
iOS native development	$9,000	$11,000	$37,000	$45,000	$60,000	$90,000
Android native development	$9,000	$11,000	$37,000	$45,000	$60,000	$90,000
Back-end Programing, Database and APIs	$5,000	$7,000	$9,000	$10,000	$12,000	$12,000
Application Administration	N/A	$7,500	$9,000	$15,000	$10,000	$20,000
Quality Assurance and Refinements iOS	$2,000	$2,500	$5,000	$6,500	$7,000	$9,000
Quality Assurance and Refinements Android	$2,000	$2,500	$5,000	$6,500	$7,000	$9,000
Deployment and Cloud	$1,500	$2,000	$2,000	$3,000	$2,000	$3,000
Total (One Platform)	$21,000	$35,500	$68,000	$81,000	$101,000	$152,000
Total (Android and iPhone)	$32,000	$49,000	$110,000	$125,500	$168,000	$251,000

Figure 3 Cost Breakdown (Kohan, 2015)

Budgeting for the revamp of the Dayforce HCM application will primarily be focused on the salary of the six key members who will be carrying out the necessary task to make this project a success.

As seen in the Figure 3 above (Kohan, 2015), there is a different price points for different size products. For this proposal we will be rebuilding the application and database for all companies that use the Dayforce HCM system and that will be a large enterprise project. The table above is also showing the complete cost to build an app from start to end, but we will only be using a few team members as outlined in red.

Table 6 Actual Cost Breakdown

Position	Cost(Yearly)	# Of Weeks Needed	Total Cost
Android Developer	$90,000	7	$12,115
IOS Developer	$90,000	7	$12,115
Backend(Database) Developer	$12,000	3	$3,000
Application Administrator	$20,000	1 ½	$2,500
Android QA tester	$9,000	2 ½	$1,875
IOS QA tester	$9,000	2 ½	$1,875

As the new table above shows, using only the members that we need the total cost can be dropped drastically. The total cost for the employees will be $33,480 for rebuilding both the android and ios app.

Conclusion

As mentioned in the introduction, the Dayforce HCM app used for scheduling at Sportchek retail stores is lacking in quite a few areas. The slow schedule updates, lack of shift trading options and lack of information needed for shifts creates a incohesive experience.

With the restructuring of the availability tab along with the expansion on the available schedule view, this update will create a more pleasurable experience for employees, as well as management. Employees will be able to access more of their schedules and management will have an easier experience when creating schedules if the availability is filled out properly.

In addition, with the added feature of shift trading employees will be much more pleased in the ease of getting a shift covered and management will save valuable time, not having to manually change the schedule in the database each time an employee request to have their shift changes.

Finally, the added feature of showing event information for certain shifts would allow employees to be better prepared for their upcoming shifts and will create a more prepared and reliable work force. With the addition of all of these features, management will save time, employees will be better prepared, and the company will have a better working team.

Recommendation

The update of this app has no needed start date, but we recommend starting as soon as possible, allowing users to start using the new features sooner rather than later. The execution of this proposal will allow employees to have a simpler and more enjoyable experience, while providing managers more time to carry out important task, instead of having to worry about restructuring data for shift changes.

While to cost may be initially high, the fact is that Dayforce is used by many more company's than just Sportchek, and this update would benefit all parties involved and may bring more business to the Dayforce company.

References

Kohan, B. (2015, February 24). *Mobile App Development Cost and Process*. Retrieved from
http://www.comentum.com: http://www.comentum.com/mobile-app-development-cost.html

Kovačević, G. (2016, January 12). *How Long Does It Take To Build An App*. Retrieved from
https://infinum.co: https://infinum.co/the-capsized-eight/how-long-does-it-take-to-build-an-
app

Kristof, K. (2011, April 5). *How to build an iPhone app in 6 easy steps*. Retrieved from
https://www.cbsnews.com: https://www.cbsnews.com/news/how-to-build-an-iphone-app-in-6-
easy-steps/

Meenakshi, A. (2017, August 22). *Building REST API in the blink of an eye! (MySQL + NodeJS)*. Retrieved
from https://medium.com: https://medium.com/@avanthikameenakshi/building-restful-api-
with-nodejs-and-mysql-in-10-min-ff740043d4be