1. 인사  
   Hello everyone, this is report presentation for application ***“ADDON”*** by ***“GOODTEAM”***
2. 아이디어  
    For idea, we thought it would be great if there’s an application to help self-employed manage the workers schedule easily. we wanted to provide functions not only for just date schedule, but also additional functions such as attendance check.
3. 개발 환경  
    For the develop environments, we organize few number of modules distinguished by the purpose which is easy to construct.   
    For the server, Oracle server and Spring boot was chosen because we conclude that for managing various types of data objects, it will be more efficient to use sql based database than no-sql like firebase.   
    And we used android studio.
4. 서버 구조  
    Next, server and Data architecture. for enabling data from application to database, we built the required components in Spring such as mapper and controller.  
    Because we figured our service is going to use several data objects for various functions, we organize few data objects which is represented as necessity. Branch for manager, user information, works for each date tasks, etc.
5. 실 구현  
    In the implementation process, we firstly use Navigation and Data binding to build connection between each fragments and access to variables without call views by Id.  
    And to build up connection gate between server and software, we used Retrofit API, which provides ways to be managed easily with assigning separately by our needs.
6. 실 구현  
    For utilizing data with list visualization, we adapted both RecyclerView and ListView. Since our service must provide considerable number of lists, it can be treated as one of the components which we used the most. They were used in each fit environment, distinguished by whether list is static or dynamic.   
    About the CalendarView, one of the main functions of our service is providing schedule, it was inevitable to use. But we noticed that the original calendar doesn’t provide every function we needed, so we adapt the improved calendar called materialCalendarView from outside and customize it for our functions.
7. 실 구현  
    Lastly, TimePicker was used for functions with selecting the commitment time and leaving time. And Dialog was implemented in many places to support functions that make user can modify some information such as each payment, time and even select the branch or worker they want to see information.
8. 결과  
    As the result, for the basic, the main page to log in and signup page for creating new account has been built first.   
    Even though we built the boss pages first, since the worker page’s volume is lot smaller than boss, we will introduce the worker page first.  
    And for the worker users, schedule page which provides the summary information was built. In this page, worker can check their work schedule for each branch, attendance status and the estimated payment for their works.
9. 결과  
    In side of Boss user, first, history page. In this page, user can observe the date schedule with timetable for all workers by clicking the date in calendar and summary for each workers which contains information with modifying attendance, checking whether the worker is full time and night time. Of course, they can experience all the same thing by each branch too.  
    Next page is schedule page. In here, boss user can modify each worker’s schedule in detail with date, time and payment for hour. As the same, this function is also available by all branches.  
    Last one is management page. In this page, user can add, modify or delete branches and workers as they want.
10. 결론  
     As the conclusion, we evaluate ourselves that the coding skills are quite improved through this project and we succeeded to implement most of functions we aimed from the start.  
     But, there were more feedbacks to make improvement. We underestimate and misdesign the data structure, which cause time waste on coding and the delay of application. Also, there was bit regret to study harder, then maybe we could make a more organized application. Thank you for listening.