**모바일/웹서비스 프로젝트 공통평가 01**

**수행 결과 보고서**

**학번 : 2019102624**

**이름 : 김진환**

[적/부판정 범례] ○:적합, ×:부적합, △:점검제외, N/A:해당없음

첨부 그림은 모두 문서에 포함하는 형식으로 저장 할 것

| **수업 요구사항** | | | | **이행현황** | | |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **구분** | **요구사항명** | **요구사항 내용** | **완료**  **여부** | **적/부 판정** | **관련증빙** |
| 1 | 제출문 | 폴더 구조 | - ROOT  - PhotoBlogServer[Dir]  : Django Server  - PhotoViewer[Dir]  : Android Client  -모바일/웹서비스 프로젝트 공통평가 01\_수행 결과 보고서.docx[File]  제출 : 탐색기, 파인더 또는 Git레포지토리 폴더 구조 캡처 화면 | O |  |  |
| 2 | 제출문 | Github 레파지토리 push | 아래 평가 항목에 대하여 구현 혹은 테스트를 완료하고, 캡처 이미지를 제공된 MS-Word 파일로 정리하여 Github URL을 제출 한다.  제출 : URL | O |  | https://github.com/sodlfmag/MW\_Proj\_mid |
| 3 | 강의 자료 | 포토 블로그 – API | 강의 자료 [7페이지, POST]  제출 : 1. 작성 완료 웹페이지 작성 화면 캡처  2. 게시 완료된 페이지 화면 캡처 | O |  |  |
| 4 | 강의 자료 | 포토 블로그 – API 검증(2/3) | 강의 자료 [9페이지]  제출 : 터미널 화면 캡처 | O |  |  |
| 5 | 강의 자료 | 포토 블로그 – API 검증(3/3) | 강의 자료 [10페이지]  제출 : [codebeautify](https://codebeautify.org/jsonviewer) 서비스 테스트 화면 캡처 | O |  |  |
| 6 | 강의 자료 | Server/Client 연동 테스트 [Sockec Server 요청 메시지] | 강의 자료 [24페이지]  제출 : 수신 파일 VSCODE 편집화면캡처 | O |  |  |
| 7 | 강의 자료 | Server/Client 연동 테스트 [Server 브라우저 출력화면] | 강의 자료 [25페이지]  제출 : Server 브라우저 출력화면 (localhost) | O |  |  |
| 8 | 강의 자료 | Server/Client 연동 테스트 [Client 시작 화면] | 강의 자료 [25페이지]  제출 : Client 시작 화면 | O |  |  |
| 9 | 강의 자료 | Server/Client 연동 테스트 [Client 동기화 완료 화면] | 강의 자료 [25페이지]  제출 : Client 동기화 완료 화면(localhost 서버연동) | O |  |  |
| 10 | 강의 자료 | MainActivity 클래스 [Upload 구현, 19page] | MainActivity 클래스 [Upload 구현, 19page]  1. Upload (Hard cording) : 중  (1. 2. 중 하나만 제출 ) | X |  |  |
| MainActivity 클래스 [Upload 구현, 19page]  2. Upload (Post에 따른 UI제공) : 상  (1. 2. 중 하나만 제출 | O |  |  |
| 11 |  | Pythonanywhere 클라우드상 동작 | Pythonanywhere 클라우드상 서버 동작  제출 : Server 브라우저 출력화면 (클라우드 서버연동) | O |  |  |
| 12 |  | Pythonanywhere 클라우드상 동작 | Pythonanywhere 클라우드상 서버 동작  제출 : Client 동기화 완료 화면 (클라우드 서버연동) | O |  |  |
| 13 |  | 기타 기획 되지 않은 Client 편의 기능 [최대 5개] | 제출 : 해당 기능 정의, 설명 및 동작 증빙 화면 캡처  (서버/클라이언트 각각 구현이 필요한 경우 별도 요구사항으로 등록 함) | O |  |  |
| 13-1 |  | 로딩 인디케이터 기능 | - 이미지 다운로드 중: 동기화 버튼 터치 시 ProgressBar가 표시되고 "이미지를 불러오는 중..." 텍스트가 표시됨  - 이미지 업로드 중: 새로운 이미지 게시 버튼으로 업로드 시 ProgressBar가 표시되고 "업로드 중..." 텍스트가 표시됨  - 작업 완료 시 ProgressBar가 자동으로 사라짐  1. 다운로드 시: 동기화 버튼 클릭 → ProgressBar 표시 → 완료 후 자동 숨김  2. 업로드 시: 새로운 이미지 게시 → 이미지 선택 → 제목/내용 입력 → 업로드 터치 → ProgressBar 표시 → 완료 후 자동 숨김 | O |  |  |
| 13-2 |  | Pull-to-Refresh 기능 | - RecyclerView에서 아래로 당겨서 이미지 목록을 새로고침할 수 있음  - 당기는 동안 새로고침 인디케이터(원형 애니메이션)가 표시됨  - 동기화 버튼 터치 없이 간편하게 새로고침 가능  사용 방법  1. 메인 화면에서 RecyclerView 영역을 아래로 당김(손가락으로 아래로 스와이프)  2. 새로고침 인디케이터가 표시되면서 자동으로 이미지 목록 재로드  3. 완료 시 인디케이터가 자동으로 사라짐 | O |  |  |
| 13-3 |  | 이미지 상세 보기 기능 | - RecyclerView의 이미지 항목을 클릭하면 전체 화면으로 확대된 상세 화면으로 이동  - 상세 화면에서 이미지, 제목, 텍스트 내용을 확인할 수 있음  - 검은색 배경에 이미지가 중앙에 표시되고, 하단에 흰색 패널에 제목과 텍스트 표시  사용 방법  1. 메인 화면에서 RecyclerView에 표시된 이미지 중 하나를 터치  2. ImageDetailActivity로 전환되어 전체 화면 이미지와 제목/텍스트 확인  3. 뒤로가기 버튼으로 메인 화면으로 복귀 | O |  |  |
| 13-4 |  |  | - 상세 화면에서 "갤러리에 저장" 버튼을 클릭하면 기기의 갤러리(사진 앨범)에 이미지가 저장됨  - 저장 성공 시 "갤러리에 저장되었습니다" Toast 메시지 표시  - 저장된 이미지는 기기의 갤러리 앱에서 PhotoBlog 폴더로 확인 가능  사용 방법  1. 상세 화면으로 이동 (이미지 클릭)  2. 하단의 "갤러리에 저장" 버튼 터치  3. Toast 메시지로 저장 완료 확인 | O |  |  |
| 13-5 |  | 이미지 공유 기능 | - 상세 화면에서 "공유" 버튼을 클릭하면 다른 앱으로 이미지를 공유할 수 있음  - 공유 선택 다이얼로그에서 카카오톡, 이메일, 메시지 등 다양한 앱 선택 가능  - 제목과 텍스트도 함께 공유됨  사용 방법  1. 상세 화면으로 이동 이미지 터치  2. 하단의 공유 버튼 클릭  3. 공유 선택 다이얼로그에서 원하는 앱 선택 (예: 카카오톡, Gmail, 메시지 등)  4. 선택한 앱으로 이미지와 텍스트가 전달됨 | O |  |  |