

Bridge Talk 최종 보고서

시간표 관리 · 일정 조율 · 약속 투표를 통합한 커뮤니케이션 기반 서비스 기획안

개발 스터디 프로젝트

본 문서는 Bridge Talk 프로젝트의 문제 정의, 차별성, 핵심 기능, 개발 우선순위, 역할 분배, 실행 방향을 종합 정리한 최종 보고서다.

작성일: 2026-04-19

1. 프로젝트 개요

Bridge Talk는 시간표 관리, 일정 조율, 약속 투표 기능을 하나로 묶어 사용자들이 더 빠르고 편하게 약속을 정할 수 있도록 돕는 커뮤니케이션 기반 서비스 프로젝트다. 현재 기준으로 이 프로젝트의 핵심 방향성은 단순한 일정 기록 도구를 만드는 데 있지 않고, 사람들이 실제로 가장 불편해하는 “언제 만날지 정하는 과정”을 줄이는 데 있다.

기존의 일정 관리 방식은 개인 시간표 확인, 가능한 시간 파악, 단체 대화방에서 일정 제안, 응답 수집, 최종 확정이라는 여러 단계를 거친다. 이 과정은 반복적이며, 특히 여러 명이 함께 움직이는 상황에서 비효율이 크게 나타나는 경향이 있다. Bridge Talk는 이러한 문제를 하나의 흐름으로 통합해 사용자의 조율 부담을 낮추는 것을 목표로 한다.

본 프로젝트는 학습 목적을 포함한 스터디형 개발 프로젝트이지만, 특정 조건에서 실제 생활 문제를 해결할 수 있는 베타 솔루션까지 도달하는 것을 목표로 삼는다. 따라서 기획 중심 프로젝트가 아니라, 실행 가능한 최소 기능 제품을 우선 완성하는 방식이 적절하다.

2. 문제 정의

문제의 핵심은 “일정을 관리하는 것”보다 “여러 사람의 시간을 맞추는 것”에 있다. 왜 이 지점이 중요한가 하면, 개인 일정 앱은 많아도 집단 조율 과정까지 자연스럽게 연결해 주는 서비스는 체감상 부족하기 때문이다.

현재 사용자가 약속을 잡을 때 겪는 대표적 불편은 다음과 같다.

- 자신의 시간표와 상대의 시간표를 동시에 고려해야 한다.
- 단체 채팅방에서 가능한 시간을 일일이 묻고 답하는 과정이 길어진다.
- 누가 응답했는지, 어떤 시간이 다수 의견인지 정리하기 어렵다.
- 일정 확정 이후에도 변경 사항을 다시 공유해야 한다.

이 문제는 학교 팀플, 스터디, 동아리, 소규모 모임처럼 반복적인 약속 조율이 필요한 환경에서 더 크게 나타나는 경향이 있다. 따라서 Bridge Talk는 단순 편의 기능이 아니라, 반복되는 커뮤니케이션 비용을 줄이는 서비스로 해석할 수 있다.

3. 프로젝트 목표

Bridge Talk의 목표는 크게 네 가지로 정리할 수 있다.

- 스터디 및 소규모 모임에서 일정을 쉽고 빠르게 정하도록 돕는 것이다.
- 일정 생성과 공유 과정을 단순화해 사용자의 입력 부담을 줄이는 것이다.
- 정해진 시간표를 바탕으로 가능한 약속 시간을 자동 또는 반자동으로 제안하는 구조를 마련하는 것이다.
- 사용자의 생활 흐름 속에서 실제로 사용할 수 있는 조율 도구를 만드는 것이다.

임시 결론으로 보면, 이 프로젝트의 성공 기준은 기능 수의 많고 적음보다 “약속 잡는 시간이 실제로 줄었는가”에 더 가깝다. 조건을 붙이면, 초기 버전에서는 완전한 플랫폼보다 빠르게 작동하는 핵심 흐름 구현이 더 중요하다.

4. 핵심 제안 내용

Bridge Talk의 제안은 크게 두 축으로 구성된다.

첫 번째 축은 빠른 실행이다. 최소 1주일에 한 번 작업을 진행하는 방식은 프로젝트가 장기 기획 상태에서 멈추지 않도록 하는 장치로 볼 수 있다. 학생 프로젝트에서 가장 큰 실패 요인 중 하나는 기획 과잉과 실행 부족인데, Bridge Talk는 이를 피해야 한다.

두 번째 축은 서비스 완성도의 방향 설정이다. 완벽한 기능 집합을 처음부터 모두 넣기보다, 실제 사용성이 높은 기능을 먼저 묶어 베타를 만들고 이후 개선하는 방식이 적합하다. 현재 기준으로 이 프로젝트는 고도화된 대형 플랫폼보다는 “작지만 실제로 쓸 수 있는 일정 조율 도구” 쪽이 현실성이 높다.

5. 기대 효과

5-1. 메신저 속의 일정 관리

Bridge Talk는 일정 조율이 채팅과 분리되지 않도록 설계될 가능성이 높다. 사용자는 보통 메신저에서 약속을 논의하므로, 일정 관리가 별도 앱에서만 이루어지면 다시 메시지로 돌아와 내용을 전달해야 한다. 반면 Bridge Talk는 커뮤니케이션 맥락 안에서 시간표 확인과 약속 조율을 이어 붙이는 방향성을 가진다.

5-2. 약속 투표 기능

다수 인원이 참여하는 약속에서는 특정 시간 후보를 두고 투표를 진행하는 것이 가장 단순한 방식인 경우가 많다. Bridge Talk의 약속 투표 기능은 단순 투표 그 자체보다, 일정 조율 과정을 구조화한다는 점에서 의미가 있다. 누가 가능한지, 어떤 시간이 가장 적합한지 빠르게 파악할 수 있기 때문이다.

5-3. 맞춤형 추천 기능

사용자의 시간표나 가능한 시간대를 기반으로 약속 후보를 추천하는 기능은 Bridge Talk의 실질적 차별 요소가 될 가능성이 있다. 단순히 “시간을 입력받는 앱”이 아니라, 사용자의 일정 데이터로 조율 자체를 보조하는 구조이기 때문이다. 다만 초기 버전에서는 완전 자동 추천보다 규칙 기반 추천이 더 현실적이다.

6. 차별성 분석

Bridge Talk의 차별성은 “시간표 저장”이 아니라 “조율 흐름 통합”에 있다.

일반적인 캘린더 서비스는 개인 일정 기록에 강점이 있다. 반면 Bridge Talk는 개인 시간표 확인, 가능한 시간 선별, 집단 투표, 약속 확정까지를 하나의 흐름으로 연결하려는 프로젝트다. 이는 기존 일정 앱과는 문제를 바라보는 출발점이 다르다고 해석할 수 있다.

또한 학생 환경에 맞춘 서비스라는 점도 차별화 포인트가 된다. 대학생과 스터디원들은 정해진 수업 시간, 팀플 시간, 동아리 활동, 개인 일정이 혼재하는 경우가 많다. 이런 환경에서는 단순 일정 기록보다 “빈

시간대 찾기”가 더 중요한 경우가 많다. Bridge Talk는 바로 이 지점을 겨냥한다.

현재 기준으로 이 프로젝트의 차별성은 다음과 같이 정리할 수 있다.

- 개인 관리가 아니라 집단 조율 중심 구조를 가진다.
- 채팅형 커뮤니케이션 맥락과 연결되기 쉽다.
- 시간표 데이터를 약속 추천에 활용할 수 있다.
- 학생 프로젝트 규모에 맞게 베타 기능부터 빠르게 검증할 수 있다.

7. 핵심 기능 정의

최소 기능 기준으로 Bridge Talk에 우선 필요한 기능은 아래와 같다.

7-1. 시간표 등록 및 확인

사용자가 자신의 수업, 고정 일정, 가능한 시간대를 입력하거나 확인할 수 있어야 한다. 이는 모든 추천과 조율 기능의 기반이 된다.

7-2. 약속 생성

사용자가 모임 이름, 참여 인원, 희망 날짜 범위 등을 입력해 약속 조율 방을 만들 수 있어야 한다.

7-3. 가능 시간 선택

참여자 자신이 가능한 시간대를 표시할 수 있어야 한다. 이 기능이 없으면 집단 조율이 수작업 채팅으로 다시 돌아간다.

7-4. 약속 투표

여러 후보 시간 중 선호 시간에 투표할 수 있어야 한다. 이는 다수결 또는 우선순위 선택 구조로 확장 가능하다.

7-5. 최종 일정 확정 및 공유

가장 많은 응답을 받은 시간을 최종 일정으로 확정하고, 참여자에게 결과를 전달할 수 있어야 한다.

7-6. 기본 추천 기능

등록된 시간표를 바탕으로 겹치지 않는 시간대를 추천하는 기능이 필요하다. 다만 초기 버전에서는 복잡한 AI보다 조건 기반 추천이 적합하다.

8. 개발 우선순위

문제를 실제로 해결하려면 무엇을 먼저 만들어야 하는가가 중요하다. 왜냐하면 학생 프로젝트에서 모든 기능을 동시에 잡으면 개발이 분산되고, 결국 아무것도 완성되지 않을 가능성이 높기 때문이다.

우선순위는 다음과 같이 잡는 것이 적절하다.

- 1순위: 일정 조율의 최소 흐름 완성 — 시간표 입력, 약속 생성, 가능 시간 표시, 투표, 일정 확정

- 2순위: 사용성 향상 — UI 개선, 결과 공유 방식 개선, 반복 입력 최소화
- 3순위: 추천 기능 고도화 — 시간표 기반 자동 후보 제안, 조건 필터, 우선순위 반영
- 4순위: 확장 기능 — 알림, 외부 플랫폼 연동, 사용자 그룹 관리, 기록 추적

임시 결론으로 보면, 베타 단계에서는 1순위 기능만 제대로 작동해도 프로젝트의 목적을 상당 부분 달성할 가능성이 높다.

9. 개발 방식 제안

현재 스터디 상황과 자원 조건을 고려하면, Bridge Talk는 처음부터 정식 웹서비스 전체를 만드는 방식보다 가벼운 베타를 먼저 만드는 편이 현실적이다.

특정 조건에서 가장 적합한 초기 형태는 다음 둘 중 하나다.

- 디스코드 봇 기반 프로토타입
- 간단한 웹 기반 MVP

디스코드 봇은 스터디원들이 이미 사용하는 플랫폼 위에서 테스트하기 쉽고, 인프라 부담이 상대적으로 낮다. 웹 기반 MVP는 향후 확장성은 좋지만, 인증, 데이터 저장, 배포, UI 구현까지 고려할 요소가 더 많다. 현재 기준으로 학습과 실행 속도를 함께 보려면 디스코드 봇형 베타가 더 유리한 경향이 있다.

10. 운영 및 기술적 한계

이 프로젝트의 가장 큰 현실적 제약은 운영 자금과 인프라다. 특히 사용자 데이터를 다뤄야 하는 서비스는 저장 구조와 관리 방식이 중요해진다. 하지만 예산이 거의 없는 상황에서는 대규모 데이터베이스 설계와 상시 운영까지 바로 들어가기 어렵다.

이 문제에 대한 해석은 두 가지가 가능하다.

- 현재 규모에서는 과도한 인프라를 만들 필요가 없다는 해석이다. 초기 사용자 수가 적다면 경량 데이터 저장 구조로도 충분할 수 있다.
- 장기적으로 서비스가 성장할 경우 데이터 구조를 처음부터 어느 정도 정리해야 한다는 해석이다.

임시 결론은 “초기에는 가볍게 시작하되, 데이터 구조는 확장 가능하게 설계해야 한다”에 가깝다. 예를 들어 로컬 파일, 무료 티어 DB, 소규모 클라우드 환경 등으로 시작할 수 있으나, 추후 이전 가능한 구조를 염두에 두어야 한다.

또한 다음과 같은 리스크가 존재한다.

- 기획만 길어지고 구현이 지연될 수 있다.
- 팀 내 역할이 모호해지면 참여도가 떨어질 수 있다.
- 기능 욕심이 커지면 핵심 흐름이 완성되지 않을 수 있다.
- 운영 비용과 유지보수 부담을 과소평가할 위험이 있다.

11. 역할 분배안

스터디원 구성을 고려하면 역할은 직무 중심보다 산출물 중심으로 나누는 편이 효율적이다.

기획 파트

서비스 흐름 설계, 기능 명세 정리, 사용자 시나리오 작성, 테스트 기준 정의를 담당한다. 필요 산출물은 기능 명세서, 사용자 플로우, 화면 요구사항 문서다.

개발 파트

시간표 등록, 약속 생성, 투표, 결과 확정 등 핵심 기능 구현을 맡는다. 추가로 데이터 저장 구조, 명령 흐름, API 또는 봇 명령 처리 로직을 담당한다.

디자인 파트

화면 구조 단순화, 사용성 개선, 시각적 일관성 확보를 담당한다. 초기에는 고급 브랜딩보다 “한눈에 이해되는 화면” 설계가 더 중요하다.

PM 또는 대표 역할

일정 관리, 주간 목표 설정, 작업 분배, 진행도 점검, 회의 운영을 맡는다. 프로젝트가 멈추지 않게 하는 관리 기능이 핵심이다.

12. 단계별 로드맵

- 1단계: 문제 구체화 및 기능 확정 — 누가 어떤 상황에서 이 서비스를 쓰는지 정의하고, 베타에 넣을 기능과 제외할 기능을 나눈다.
- 2단계: 프로토타입 제작 — 디스코드 봇 또는 간단한 웹 형태로 약속 생성과 투표 흐름을 우선 구현한다.
- 3단계: 내부 테스트 — 스터디원들을 실제 사용자로 두고 약속을 잡아 보며 불편 지점을 찾는다.
- 4단계: 수정 및 개선 — 입력 방식, 응답 확인, 결과 공유 방식 등 실제 사용 중 불편했던 부분을 다듬는다.
- 5단계: 최종 정리 — 프로젝트 결과물, 시연 자료, 회고 기록, 향후 확장 방향을 정리한다.

13. 종합 결론

Bridge Talk는 단순한 시간표 앱이 아니라, 여러 사람의 시간을 맞추는 데서 발생하는 반복적 불편을 줄이기 위한 일정 조율 서비스 프로젝트다. 현재 기준으로 이 프로젝트의 가치는 기능 수가 많은 데 있지 않고, 실제로 약속 잡는 과정을 줄이는 데 있다.

이 프로젝트는 학생 개발 스터디에 적합한 주제다. 이유는 분명하다. 문제 상황이 익숙하고, 사용자 확보가 비교적 쉽고, 핵심 기능이 명확하며, 기획·개발·디자인이 모두 참여할 수 있기 때문이다. 다만 성공 여부는 아이디어의 참신함보다 실행 속도와 범위 통제에 더 크게 좌우되는 경향이 있다.

따라서 Bridge Talk의 최종 전략은 다음과 같이 정리된다.

- 완성형 대형 서비스보다, 실제로 작동하는 베타를 먼저 만든다.
- 일정 조율의 핵심 흐름을 우선 구현한다.
- 학생 사용자 환경에 맞춘 사용성을 검증한다.
- 이후 필요에 따라 추천 기능과 확장 기능을 추가한다.

결국 Bridge Talk는 “많은 것을 하는 서비스”보다 “한 가지 문제를 확실히 풀어 주는 서비스”로 시작할 때 가장 성공 가능성이 높다.

14. 실행 제안

- 이번 주 안에 베타 필수 기능 5개를 확정하고 역할을 재분배한다.
- 다음 개발 목표는 웹 전체 설계보다 디스코드 봇 또는 간단한 MVP 1차 구현으로 잡는다.